Robust Audio Segmentation

presentée a la Facultè des sciences et techniques de l'ingènieur

École Polytechnique Fédérale de Lausanne

pour l'obtention du grade de docteur ès sciences

par

JITENDRA AJMERA

Bachelor of Technology in Electrical and Electronics Engineering,

Indian Institute of Technology Bombay, Mumbai, India

Thesis committee members

Prof. Juan Mosig, EPFL, Switzerland

Prof. Hervé Bourlard, directeur de thèse, IDIAP/EPFL , Switzerland

Prof. Christian Wellekens , Eurecom, France

Prof. Dan Ellis, Columbia University, USA

Dr. Andrzej Drygajlo, EPFL, Switzerland

Dr. Iain McCowan, IDIAP, Switzerland

Lausanne, EPFL

2004

Abstract

Audio segmentation, in general, is the task of segmenting a continuous audio stream in terms of acoustically homogenous regions, where the rule of homogeneity depends on the task. This thesis aims at developing and investigating efficient, robust and unsupervised techniques for three important tasks related to audio segmentation, namely speech/music segmentation, speaker change detection and speaker clustering.

The speech/music segmentation technique proposed in this thesis is based on the functioning of a HMM/ANN hybrid ASR system where an MLP estimates the posterior probabilities of different phonemes. These probabilities exhibit a particular pattern when the input is a speech signal. This pattern is captured in the form of feature vectors, which are then integrated in a HMM framework. The technique thus segments the audio data in terms of *recognizable* and *non-recognizable* segments. The efficiency of the proposed technique is demonstrated by a number of experiments conducted on broadcast news data exhibiting real-life scenarios (different speech and music styles, overlapping speech and music, non-speech sounds other than music, etc.).

A novel distance metric is proposed in this thesis for the purpose of finding speaker segment boundaries (speaker change detection). The proposed metric can be seen as special case of Log Likelihood Ratio (LLR) or Bayesian Information Criterion (BIC), where the number of parameters in the two models (or hypotheses) is forced to be equal. However, the advantage of the proposed metric over LLR, BIC and other metric based approaches is that it achieves comparable performance without requiring an adjustable threshold/penalty term, hence also eliminating the need for a development dataset.

Speaker clustering is the task of unsupervised classification of the audio data in terms of speakers. For this purpose, a novel HMM based agglomerative clustering algorithm is proposed where,

starting from a large number of clusters, *closest* clusters are merged in an iterative process. A novel merging criterion is proposed for this purpose, which does not require an adjustable threshold value and hence the stopping criterion is also automatically met when there are no more clusters left for merging. The efficiency of the proposed algorithm is demonstrated with various experiments on broadcast news data and it is shown that the proposed criterion outperforms the use of LLR, when LLR is used with an optimal threshold value.

These tasks obviously play an important role in the pre-processing stages of ASR. For example, correctly identifying *non-recognizable* segments in the audio stream and excluding them from recognition saves computation time in ASR and results in more meaningful transcriptions. Moreover, researchers have clearly shown the positive impact of further clustering of identified speech segments in terms of speakers (speaker clustering) on the transcription accuracy. However, we note that this processing has various other interesting and practical applications. For example, this provides characteristic information about the data (metadata), which is useful for the indexing of audio documents. One such application is investigated in this thesis which extracts this metadata and combines it with the ASR output, resulting in Rich Transcription (RT) which is much easier to understand for an end-user. In a further application, speaker clustering was combined with precise location information available in scenarios like smart meeting rooms to segment the meeting recordings jointly in terms of speakers and their locations in a meeting room. This is useful for automatic meeting summarization as it enables answering of questions like “who is speaking and where”. This could be used to access, for example, a specific presentation made by a particular speaker or all the speech segments belonging to a particular speaker.

Version abrégée

"Segmentation audio" signifie, de facon générale, découper un flux audio continu en régions "homogènes" du point de vue acoustique. L'homogénéité est définie en fonction de la tâche à accomplir. Cette thèse développe des techniques non-supervisées et robustes, pour les tâches suivantes : segmentation parole/musique, detection des changements d'orateur et groupement par personne[1].

La technique de segmentation parole/musique définie dans cette thèse se base sur système hybride HMM/ANN, où un MLP estime les probabilités postérieures des diffrents phonèmes. Ces probabilités ont une variation particulières au cours du temps. Cette variation est intégrée dans un système HMM. Le résultat est une segmentation en terme de segments *reconnaissables* et *non-reconnaissables*. De nombreuses expériences démontrent l'efficacité de cette technique, sur des données "broadcast news". Ces données ont de nombreuses caractéristiques réalistes : diffrents styles de paroles et de musiques, parole et musique simultanément, autres sons que parole et musique, etc.

Cette thèse propose une nouvelle mesure de distance pour trouver les changements d'orateur. Cette mesure peut être considérée comme un cas particulier de "Log Likelihood Ratio" (LLR) ou "Bayesian Information Criterion" (BIC). L'idée fondamentale est de contraindre le nombre de paramètres comme étant constant, dans les deux modèles ou hypothéses. L'avantage majeur de notre mesure comparée aux mesures LLR, BIC ou autres est qu'elle permet d'obtenir des résultats comparables *sans* qu'il faille ajuster un seuil ou pénalité. En utilisant cette nouvelle technique, il n'est plus nécessaire d'avoir une base de données de développement.

Le groupement par personnes est une tâche de classification non supervisée des donnes audio en terme de personne. Dans ce but, nous proposons une nouvelle méthode agglomérative utilisant

[1] "speaker clustering" en Anglais.

les HMM. L'initialisation se fait avec un grand nombre de groupes, les groupes *les plus proches* sont fusionnés de facon itérative. Un nouveau critère de fusion est proposé, ne nécessitant pas d'ajuster une valeur de seuil. Le critère d'arrêt est ainsi automatique : l'algorithme s'arrête lorsqu'il ne trouve plus une paire de groupes pouvant être fusionnés. Les expériences sur des enregistrements "broadcast news" montrent que ce nouveau critère donne de meilleurs résultats que LLR, lorsque LLR est utilisé avec une valeur de seuil optimal.

Ces trois modules ont un rôle important dans les étapes précédant la reconnaissance de parole automatique (ASR). Par exemple, identifier correctement les segments *non-reconnaissables* dans le flux audio permet de les exclure de l'ASR. Ceci permet de gagner beaucoup de puissance de calcul, qui serait autrement gaspillée. D'autre part, sur les parties classifiées comme *reconnaissables*, le texte résultant a une meilleure qualité. D'autre part, des chercheurs ont clairement montré l'impact positif de l'identification en terme de personne (groupement par personne) sur la qualité de la transcription automatique. D'autres applications existent : par exemple, les trois modules fournissent de facon automatique des annotations ou métadonnées, ce qui est très utile pour l'indexation de documents. Dans cette thèse, nous présentons un exemple d'application qui extrait les métadonnées et les insèrent dans le texte produit par l'ASR. Le résultat est une Transcription Riche (RT), qui est lisible et facile à comprendre pour un utilisateur. Dans une autre application, le groupement par personne est combiné avec une information de position spatiale dans une salle de réunion. L'information résultante "qui parle où ?" est très utile pour la production automatique de notes de réunion. Par exemple, il est ainsi possible pour un utilisateur qui n'aurait pas été présent dans une réunion d'aller écouter directement la présentation faite par cette personne.

Contents

1 Introduction **1**

1.1 Contributions of this thesis . . . 3

1.1.1 Speech/Music Segmentation . . . 3

1.1.2 Speaker change detection . . . 4

1.1.3 Speaker Clustering . . . 5

1.2 Organization . . . 6

2 Background Information and Previous Work **9**

2.1 Background Information . . . 9

2.2 Previous Works . . . 15

2.2.1 Speech/Music Segmentation . . . 15

2.2.2 Speaker Change Detection . . . 17

2.2.3 Speaker Clustering . . . 19

3 Speech/Music Segmentation **23**

3.1 Introduction . . . 23

3.2 Posterior probability based features . . . 26

3.2.1 Entropy . . . 26

3.2.2 Dynamism . . . 27

3.3 Speech/Music Segmentation . . . 28

3.3.1 HMM Classifier . . . 30

3.4 Evaluation Experiments . . . 31

3.4.1 Implementation 31
3.4.2 Evaluation 32
3.4.3 Results 32
3.4.4 Discussion 35
3.5 Extension and Application to ASR 37
3.5.1 Confidence Measure 37
3.5.2 Relation to ASR Recognition Accuracy 38
3.6 Conclusion 40

4 Speaker Change Detection 43

4.1 Introduction 43
4.2 Formulation of the problem 44
4.3 Common criteria 45
4.3.1 Log Likelihood Ratio (LLR) 45
4.3.2 Bayesian Information Criterion (BIC) 46
4.4 Proposed Criterion 47
4.5 Experimental Setup 51
4.5.1 Evaluation Metric 52
4.5.2 Results 52
4.6 Conclusion 55

5 Speaker Clustering and Segmentation 57

5.1 Introduction 57
5.2 Speaker Clustering Algorithm 60
5.2.1 Initialization 60
5.2.2 Segmentation 62
5.2.3 Cluster Models Re-Training 63
5.2.4 Cluster Merging 63
5.3 Merging and Stopping Criteria 65
5.3.1 Thresholded Log Likelihood Ratio 65
5.3.2 Log Likelihood Ratio and Proposed Stopping Criterion 65

5.3.3 Proposed Merging and Stopping Criterion . . . 66
5.4 Evaluation Metric and Data . . . 68
5.4.1 Evaluation Metric . . . 69
5.4.2 Data . . . 70
5.5 System Settings and Parameters . . . 71
5.5.1 Feature Vectors . . . 71
5.5.2 Initialization . . . 72
5.5.3 Initial number of clusters (K) . . . 73
5.5.4 Initial number of Gaussians per cluster (M) . . . 74
5.5.5 Minimum Duration (MD) . . . 76
5.6 Experiments and Analysis . . . 77
5.6.1 Discussion . . . 78
5.6.2 Comparison of the proposed criterion with LLR . . . 79
5.7 Participation in NIST evaluations . . . 80
5.7.1 Evaluation Criterion . . . 80
5.7.2 Data . . . 82
5.7.3 System Parameters . . . 82
5.7.4 Comparison with other approaches . . . 83
5.8 Conclusion . . . 84

6 Applications **87**
6.1 Online Audio Indexing . . . 88
6.1.1 Segmentation . . . 88
6.1.2 Speech/Non-speech Discrimination . . . 90
6.1.3 Speaker Clustering . . . 91
6.1.4 Speech Recognition . . . 92
6.1.5 System Performance . . . 93
6.1.6 Application to Multimedia Content Analysis . . . 95
6.2 Clustering Speakers and Their Locations in Meetings . . . 96
6.2.1 Feature Extraction . . . 98

6.2.2 Clustering Locations 98
6.2.3 Combined System 99
6.2.4 Experiments and Evaluation 100
6.3 Conclusions 103

7 Conclusion 105

7.1 Speech/Music Segmentation 105
7.2 Speaker Change Detection 106
7.3 Speaker Clustering 107
7.4 Integration 108
7.5 Future Directions 109

Curriculum Vitae 123

List of Figures

1.1 The usefulness of the three tasks addressed in this thesis from an ASR point of view . 2

1.2 A screen-shot of a utility developed to segment an audio stream 6

3.1 Block diagram of the proposed speech/music segmentation system 28

3.2 Distribution of local and average entropy for speech and music 29

3.3 Distribution of local and average dynamism for speech and music. 29

3.4 HMM topology for the proposed speech/music segmentation system 30

3.5 Word error rate as a function of confidence score 39

4.1 Two neighboring windows with acoustic vector sequences X and Y around time t, when we want to decide if a change point exists or not. 45

4.2 The behavior of the proposed distance metric on a sample speech 49

4.3 The histogram of deleted true reference speaker change points as a function of their detectability 54

4.4 Histogram of the biases of the detected change points with respect to the location of the true reference change points 55

5.1 The overall description of the speaker clustering algorithm 61

5.2 HMM Topology used for speaker clustering 62

5.3 The segmentation, resulting Viterbi score and associated number of clusters after every iteration in proposed speaker clustering algorithm 68

5.4 Performance of the speaker clustering algorithm in terms of with varying number of initial clusters 75

5.5 Performance of the speaker clustering algorithm with different number of initial number of Gaussian components 76
5.6 The behavior of the proposed speaker clustering algorithm with different values of minimum duration 77
5.7 Breakdown of performance of different participating systems in RT-03 speaker diarization evaluations in terms of False Alarm Speaker Time, Missed Speaker Time and Speaker Error Time 85

6.1 Block diagram of online audio indexing system 89
6.2 IDIAP Smart Meeting Room 96
6.3 Recording setup 101
6.4 Time-line of speech activity for the 6 speakers 102

List of Tables

2.1 Summary of previous work on speaker change detection (SCD) 19

2.2 Summary of previous works done on speaker clustering 22

3.1 Speech/Music classification results for Test Set 1 32

3.2 Speech/Music Classification results for Test Set 2 33

3.3 Speech/Music classification results for Test Set 3 34

3.4 Speech/Music classification results for Test Set 4 34

3.5 Comparative study of speech/music segmentation systems using supervised and unsupervised training methodologies. The results show that both the systems perform very similarly on average. 35

4.1 Results of the proposed criterion and BIC (with different values of λ) on HUB-4 1997 evaluation data. 53

4.2 Results of the proposed criterion and BIC (with $\lambda = 6$) after ignoring multiple speaker segments, insertions made during music regions and slightly (less than 2 seconds) deviated change points. 54

5.1 Comparison of performance of MFCC, LPCC and PLPC coefficients in the proposed speaker clustering algorithm 72

5.2 Comparative clustering performance in terms of Q-Measure for K-Means and Uniform initialization 73

5.3 Speaker clustering performance in the cases when the number of speakers were known and unknown 74

5.4 Speaker clustering performance in terms of average speaker purity (asp), average cluster purity (acp) and Q-Measure 78
5.5 The comparative performance in terms of the Q-Measure of three systems differing in the employed merging and stopping criteria 79
5.6 The performance of different speaker clustering systems participated in Dryrun experiments 83
5.7 Performance of different systems participated in NIST speaker diarization evaluations 83
5.8 The analysis of the errors for the cases when a speech/non-speech discrimination (SND) was applied and when it was not applied 84

6.1 Summary of performance for different modules of the proposed online audio indexing application 95
6.2 HTER and Frame Accuracy (FA) in percentages for the acoustic only and for the combined system 102

Chapter 1

Introduction

Audio segmentation is the task of partitioning an audio stream in terms of homogeneous regions. This definition of the task is quite abstract without specifying the rule of homogeneity, which in turn depends on the application. In Automatic Speech Recognition (ASR), for example, an initial segmentation is required in terms of homogeneous speech and non-speech regions. Having segmented speech regions, it is also often necessary to segment these further in terms of homogeneous speaker turns. Speaker turn information can be helpful for speaker adaptation in ASR, rich transcription of videos, meetings etc. Hence, given an audio stream, there are many possible ways it can be segmented.

The problem of distinguishing speech signals from other non-speech (pre-dominantly music) signals is becoming increasingly important as Automatic Speech Recognition (ASR) systems are being applied to an increasing number of real-world multimedia applications. Furthermore, audio and speech segmentation will always be needed to break the continuous audio stream into manageable chunks applicable to the configuration of the ASR system. By using ASR acoustic models trained for a particular acoustic condition, such as wide bandwidth (high quality microphone input) versus telephone narrow bandwidth, male speaker versus female speaker, etc., overall performance can be significantly improved. Finally, this segmentation could also be designed to provide us with additional interesting information such as division into speaker turns and speaker identities allowing for automatic indexing and retrieval of all occurrences of the same speaker. It can also provide "syntactical information" such as end of sentences, punctuation marks etc. Moreover, speaker seg-

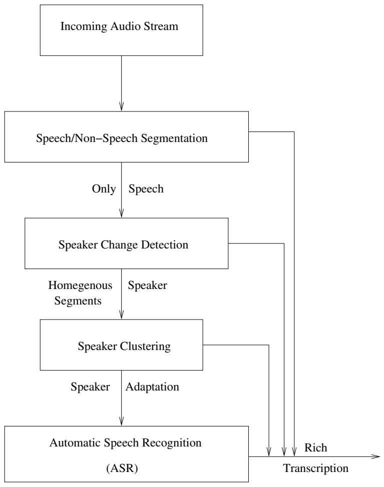

Figure 1.1. The usefulness of the three tasks addressed in this thesis from an ASR point of view. The speech/non-speech segmentation allows only speech segments to be passed to the recognizer, saving computation time as well as improving recognition accuracy. The speech segments are further segmented in terms of speakers, which is further useful for ASR as decoding of well segmented and manageable speech chunks is always more easier and accurate. These homogenous speaker segments can be further clustered together in terms of speakers to facilitate speaker adaptation of an ASR system. Finally, the output of all these modules can also be combined with the output of the ASR system, resulting in Rich Transcription.

mentation information can also be used for efficient speaker adaptation, which has been shown to significantly improve ASR accuracy (Chen and Gopalakrishnan, 1998; Johnson and Woodland, 1998; Ramabhadran *et al.*, 2003). Finally, all this information, when combined with the text output of the ASR, results in a Rich Transcription (RT) which is much easier to understand. This is further illustrated in Fig. 1.1.

In addition to improving ASR systems, audio segmentation is also useful in many other interesting practical applications. Content based audio classification and retrieval have a wide range of applications in the entertainment industry (ASSAVID Project, 2000), audio archive management, commercial music usage, surveillance, etc. There are also distributed audio libraries on the World Wide Web and audio segmentation will be needed for sound indexing and search. This is also an integral component of content-based indexing, archiving, retrieval, and on-demand delivery of audiovisual content (CIMWOS Project, 2000). Audio segmentation would also be an important tool in summarizing meetings, which has recently gained a lot of interest in the research community (M4 Project, 2002). For example, segmentation of the speech data in terms of speakers could help in efficient navigation through audio documents like meeting recordings or broadcast news archives. Using these segmentation queues, an interested user can directly access a particular segment of the speech made by a particular speaker. In situations like meeting recordings, information about presentations made by participants can be automatically and efficiently extracted.

This thesis thus aims at developing and investigating different classification techniques to segment complex audio signals (typically contained in large multimedia documents or broadcast news), and comprised of a sequence or a mix of complex signals (such as music, speech from different speakers, different acoustic environments etc.), with a view to further processing, e.g., indexing and recognition. In summary, the following three problems will be the main focus of this thesis:

1. Speech/Music Segmentation: To begin with, the audio can be broken down into speech and music regions. This segmentation can also be extended to finding situations where both speech and music are present (referred to as "mixtures" in this thesis) and then finding the proportion of speech in these mixture regions.

2. Speaker change detection: Assuming that the input signal is only speech, finding the speaker boundaries *i.e.* begin and end point for every speaker.

3. Speaker clustering: To find speaker segments with not only the correct boundaries, but also consistent identity-labels. This also involves finding the actual number of speakers in the speech data.

The next section looks at various contributions of this thesis in addressing these three problems.

1.1 Contributions of this thesis

The focus of this thesis was to develop unsupervised and robust techniques for audio segmentation. More precisely, techniques which are independent of any thresholding and which can perform robustly across a variety of data. The following are the contributions in brief, specific to the following three tasks.

1.1.1 Speech/Music Segmentation

The work in this domain was inspired by Williams and Ellis (1999). Specifically, the posterior probability based features namely "entropy" and "dynamism" are used in an HMM classification framework (Ajmera *et al.*, 2003). These posterior probabilities are the output of an Multi-Layer Perceptron (MLP) which is an integral module of a hybrid speech recognizer (Bourlard and Morgan, 1994). Specifically, in this thesis a framework is proposed for this task with the following properties:

- Unlike most of the previous work on this topic which was meant for discriminating discrete segments of speech and music, this framework is meant for segmenting a continuous audio stream.
- A detailed study about the relative performance of the two features (entropy and dynamism) and also comparison of their performance with that of standard MFCC features.
- The possibility of training the models for the two classes in a completely unsupervised manner was also explored.
- Testing this technique on data having different speech and music styles, as well as different temporal distributions of the speech and music signals.
- Extending the problem to finding "mixtures" of sounds, *i.e.* regions where both speech and music are present simultaneously (*e.g.* broadcast news). To facilitate this, a confidence measure was defined which effectively indicates the proportion of speech in an audio segment. A study of how this confidence score is an indication of the word recognition accuracy is also presented in this thesis.

1.1.2 Speaker change detection

Much of the previous work on this topic uses the **Bayesian Information Criterion (BIC)** or **Log-likelihood Ratio (LLR)** as a decision making criteria. These criteria suffer from a drawback that they involve (in some form) an adjustable threshold/penalty term. This threshold term is found empirically using a development dataset.

In this thesis, an algorithm is proposed for this task which features the following properties:

- It uses a decision making criterion which does not use a threshold/penalty term.
- The above mentioned property makes the algorithm robust to a variety of unseen data. This was tested using a database which features all kind of acoustic conditions, segment duration etc.

- A full real-time speaker change detection framework was developed which uses this decision criterion.
- The framework is completely unsupervised, *i.e.* no prior speaker models and thus no training data is required. The speaker models are created and updated dynamically.

1.1.3 Speaker Clustering

Most of the previous works on speaker clustering have limitations either in the form of assumptions about segmentation of the data or assumption about the number of speakers. Also, since most commonly used objective functions (*e.g.* likelihood or entropy) keep monotonically increasing or decreasing with variation in the number of clusters, some kind of heuristics or threshold values are necessary to make the algorithm converge at some point. Some techniques use a threshold/penalty term to select a good stopping point. These heuristics make the system less robust to unseen data and generally require an additional development dataset to adapt.

This thesis proposes a solution which features the following properties:

- No prior assumptions are made about the segmentation of the data. The segmentation and clustering are performed in an iterative manner.
- Completely unsupervised, *i.e.* no development/training data is required to create speaker models.
- A new merging criterion (for decision about merging of two clusters) is proposed, which does not use any adjustable threshold/penalty term.
- A fully automatic stopping criterion, resulting into maxima of a likelihood based objective function.

This algorithm was applied to NIST speaker diarization evaluations[1] and the performance of the algorithm was found to be highly competitive as shown later in this thesis. Apart from this, the algorithm was also applied to databases like meeting room recordings, a step toward meeting summarization.

[1] http://www.nist.gov/speech/tests/rt/rt2003/spring/

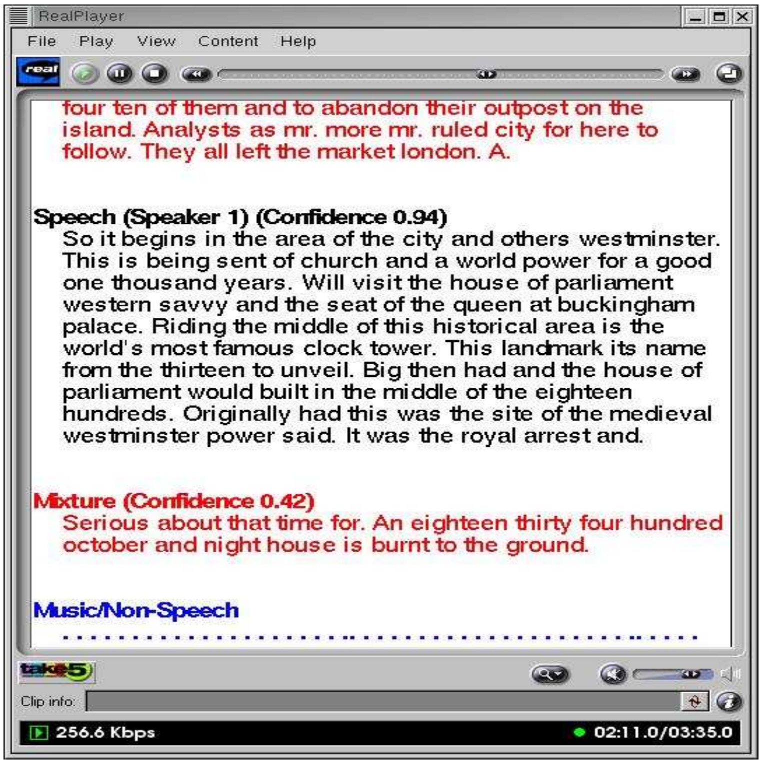

Figure 1.2. A screen-shot of a utility developed to segment an audio stream in terms of speech/music (together with "confidence" of speech), and different speaker turns and identities. Low confidence regions are marked *mixture* are not given any speaker identity. Moreover, only the mixtures and clean speech regions are recognized, saving a lot of computation time for ASR.

Apart from these specific contributions, these three tasks were also put together in an online system (McCowan *et al.*, 2003b) in the framework of (ASSAVID Project, 2000) and (CIMWOS Project, 2000). This also involved, in some cases, modification of these algorithms. Figure 1.2 illustrates the integration of various tasks related to audio segmentation into one system.

1.2 Organization

This thesis is organized as follows:

- Chapter 2 discusses the background work related to different audio segmentation problems. This includes discussion of the algorithms which are most commonly used (state-of-the-art), with their advantages and disadvantages. This is important, as much of the work proposed in this thesis is motivated by the limitations of these existing algorithms.

- In Chapter 3, a speech/music segmentation system is presented. This chapter first discusses posterior probability based features, namely entropy and dynamism, and then the HMM framework in which these features are used. This chapter also extends the work to defining a confidence measure indicating the proportion of speech in a segment.

- Chapter 4 first explains some of the most common decision making criteria for speaker change detection like BIC, LLR, Gish-distance and their limitations. This chapter then defines the proposed criterion, which does not use any adjustable threshold/penalty term, and explains a real-time framework to find speaker boundaries in a continuous audio stream.

- Chapter 5 addresses the problem of speaker clustering. It first defines the problem from the point of view of maximization of an objective function, and then presents a novel algorithm and framework achieving this. The chapter also presents the participation of this system in NIST evaluations.

- The integration of these discrete tasks into practical applications including online and real-time audio segmentation and segmentation of meeting room recordings is also explored in this thesis. Chapter 6 discusses various issues and performance associated with these applications.

Chapter 2

Background Information and Previous Work

The rapid advance in speed and capacity of computers and networks have allowed the inclusion of audio as a data type in many modern computer applications. Multimedia databases or file systems can easily have thousands of audio recordings. However, the audio is usually treated as an opaque collection of bytes with only the most primitive fields attached: name, file format, sampling rate, etc. Content based audio retrieval, thus, has become a very useful and necessary task in most audio based applications. It is because of this that audio processing has gained a lot of interest and focus recently. In this chapter (in Section 2.1) we present a summary of most commonly used statistical and signal processing tools used in state-of-the-art audio processing techniques. This chapter also discusses (in Section 2.2) most of the significant work done to date in addressing various problems related to audio segmentation, and its limitations.

2.1 Background Information

In this section, some of the state-of-the-art statistical and signal processing tools used in most audio processing techniques as well as in this thesis are briefly mentioned, mainly to define notations and abbreviations.

Acoustic Features

All audio processing techniques start by converting the raw speech signal into a sequence of acoustic feature vectors carrying characteristic information about the signal. This preprocessing module (feature extraction) is also referred to as "front-end" in the literature. The most commonly used acoustic vectors for this purpose are Mel Frequency Cepstral Coefficients (MFCC) (Davis and Mermelstein, 1980), Linear Prediction Cepstral Coefficients (LPCC) (Makhoul, 1975) and Perceptual Linear Prediction Cepstral (PLPC) Coefficients (Hermansky, 1990).

All these features are based on the spectral information derived from a short time windowed segment of speech. They differ mainly in the detail of the power spectrum representation. MFCC features are derived directly from the FFT power spectrum, whereas the LPCC and PLPC use an all-pole model to represent the smoothed spectrum. The mel-scale filterbank centers and bandwidths are fixed to follow the mel-frequency scale, giving more detail to the low frequencies. LPCC features can be considered as having adaptive detail in that the model poles move to fit the spectral peaks wherever they occur. The detail is limited mostly by the number of poles available. PLPC features are a hybrid between filterbank and all-pole model spectral representation. The spectrum is first passed through a bark-spaced trapezoidal-shaped filterbank and then fit with an all-pole model. The details of the PLPC representation is determined by both the filterbank and the all-pole model order. The spectral representation is transformed to cepstral coefficients as a final step. This is done because of the (near) orthogonalizing property of the cepstral transformation. The filterbank representation are transformed directly by a Discrete Cosine Transform (DCT). The all-pole representations are transformed using the recursive formula between prediction coefficients and cepstral coefficients (Rabiner and Juang, 1993). In all cases, discarding the zeroth cepstral coefficient results in energy normalization.

PLPC features are used in most state-of-the-art ASR systems (Matsoukas *et al.*, 2003; Woodland *et al.*, 2003). The effectiveness of LPCC features for automatic speaker identification and verification and recognition was shown in (Atal, 1974, 1976). However, MFCC features are now being used in more and more speaker recognition applications. For example, most of the participating systems in NIST speaker recognition evaluations in 1998 used MFCC features and some systems used LPCC features (Martyn *et al.*, 2000). Following the trends in many state-of-the-art speaker recognition systems (*e.g.* (Reynolds *et al.*, 2000)), MFCC coefficients (without energy term (C_0) and derivatives) are used as acoustic feature vectors in this thesis, unless otherwise mentioned. Also, PLPC features are used as a front-end for ASR in Chapter 3 and 6.

A sequence of acoustic feature vectors is denoted as $X = x_1, x_2, .., x_n, ..., x_N$ where the subscript n denotes the time-frame index and N denotes the total length of the sequence in terms of these time-frames. The time interval between two frames is usually of the order of 10-16 ms.

Hidden Markov Model (HMM)

Most of the work in this thesis is based on Hidden Markov Models (HMM) which are a highly effective means for sequence analysis (Rabiner and Juang, 1993). A HMM is a stochastic finite state machine, consisting of a set of states $q_k, 1 \leq k \leq K$, a set of emission distributions (one for each state) and corresponding transitions probabilities between states. Each state in the HMM generates an observation (physical event). When used for speech, the HMM usually has a left-to-right topology and models the sequence of acoustic feature vectors X as a piecewise stationary process.

The parameters of the HMM will be denoted as Λ in this thesis. Given a sequence of observation vectors X, the parameters of the HMM are trained using Expectation-Maximization (EM) algorithm (Dempster *et al.*, 1977), as expressed in the Baum-Welch algorithm for HMM, to maximize the likelihood, *i.e.* $\Lambda^* = \arg\max_\Lambda p(X|\Lambda)$. However, in the current work, we have used the Viterbi approximation of this algorithm, where, in an iterative process, first a best hidden state sequence (q_{best}) is derived based on the current set of HMM parameters (Λ) using the Viterbi algorithm (Viterbi, 1967), *i.e.*:

$$q_{best} = \arg\max_q p(X, q|\Lambda) = \arg\max_q p(X|q, \Lambda) \cdot p(q|\Lambda) \tag{2.1}$$

then the parameters of the HMM are updated using this best state sequence as:

$$\Lambda^* = \arg\max_\Lambda p(X, q_{best}|\Lambda) \tag{2.2}$$

Also, note that the term $p(X, q_{best}|\Lambda)$ is referred to as *Viterbi score* in this thesis.

The likelihood of an observation vector x_n given state q_k, $p(x_n|q_k)$, can be estimated using either Gaussian Mixture Model (GMM) or a Multi-Layer Perceptron (MLP) as explained below.

Gaussian Mixture Model (GMM)

A GMM is a mixture of several Gaussian distributions and is used to estimate the Probability Density Function (PDF) of a sequence of feature vectors. The likelihood of a model (GMM) given observation x_n is then estimated as:

$$p(x_n|\mathcal{N}(x_n, \mu_i, \Sigma_i)) = \sum_{i=1}^{M} \frac{w_i}{\sqrt{2\pi|\sum_i|}} \exp\{\frac{-(x_n-\mu_i)^T \sum_i^{-1}(x_n-\mu_i)}{2}\} \tag{2.3}$$

where M is the number of Gaussian distributions in the GMM. The parameters of these distributions, w_i, μ_i and $\sum_i$, are respectively the weight, mean and diagonal covariance matrix of the i^{th} distribution in the GMM. Given a sequence of observation vectors, the parameters of a GMM can be trained via the EM algorithm to maximize the likelihood of the data.

Also, the observations in X are assumed to be independent and identically distributed (i.i.d.). Accordingly, the likelihood of a model (*e.g.* a GMM or an HMM) parameterized by θ given observation sequence X is estimated as:

$$p(X|\theta) = \Pi_{n=1}^{n=N} p(x_n|\theta) \tag{2.4}$$

Multi-Layer Perceptron (MLP)

An MLP is a special case of Artificial Neural Network (ANN) with one input layer, one or several hidden layers and one output layer. The parameters of this MLP are trained using the Error Back Propagation (EBP) algorithm (Bishop, 1996). An MLP can be trained to estimate posterior probabilities of different classes. When this is used in conjunction with the HMM (*e.g.* in hybrid systems (Bourlard and Morgan, 1994)), these classes are the states of the HMM. The MLP thus estimates the posterior probability $P(q_k|x_n)$ that an observation vector x_n belongs to class q_k. Using Bayes rule, these posterior probabilities can be turned into scaled likelihoods to be used in HMM as:

$$\frac{p(x_n|q_k)}{P(x_n)} = \frac{P(q_k|x_n)}{P(q_k)} \tag{2.5}$$

where $P(q_k)$ is the prior probability of the class q_k, as estimated on the training data, and $P(x_n)$ is independent of the class and simply appears as a constant scaling factor.

Hypothesis Testing

Setting up and testing hypotheses is an essential part of statistical inference and is used in this thesis. In each problem considered, the question of interest is simplified into two competing claims/hypotheses between which we have a choice; the null hypothesis, denoted H_0, against the alternative hypothesis, denoted H_1. The null hypothesis, H_0 represents a theory that has been put forward, either because it is believed to be true or because it is to be used as a basis for argument, but has not been proved. The alternative hypothesis H_1 on the other hand, is a statement of what a statistical hypothesis test is set up to establish.

In the case of pattern recognition, the goal is to determine to which category or class a given observation (or sequence of observations) X belongs. If the PDF of each class is known, this becomes a problem in statistical hypothesis testing where H_0 and H_1 are defined as:

$$H_0: \qquad X \text{ is from class } \omega_1$$

and

$$H_1: \qquad X \text{ is from class } \omega_2$$

The optimum test according to the Bayes decision rule for minimum error to decide "not to reject" H_0 is a likelihood ratio test (Fukunaga, 1990) given by:

$$\frac{p(X|H_0)}{p(X|H_1)} \geq \frac{P(H_1)}{P(H_0)} \tag{2.6}$$

where $p(X|H_i), i = 0, 1$, is the likelihood of the data X under the hypothesis H_i and $P(H_i)$ is the prior probability of hypothesis H_i. If we assume the prior probability of the two classes to be equal, the term on the right hand side of (2.6) is equal to one. The term on the left hand side of (2.6) is referred to as a likelihood ratio. Strictly speaking, the likelihood ratio test is only optimal when the likelihood functions are known exactly. In practice this is rarely the case and the likelihood ratio test usually involves a threshold value. A decision to "reject H_0 in favor of H_1" is made only if the likelihood ratio is less than threshold.

When the PDFs of two classes (hypotheses) are Gaussian densities or Gaussian mixture densities (which is always the case in this thesis), it is more convenient to compute and write the **Log Likelihood Ratio (LLR)** rather than writing the likelihood ratio itself. The LLR is computed by taking the logarithm of the likelihood ratio and now decision rule becomes:

$$\log p(X|H_0) - \log p(X|H_1) \lessgtr threshold \tag{2.7}$$

In many situations, the problem is first formulated as a hypothesis testing problem and LLR is then used to make a decision. LLR is widely used in speaker recognition research (Gish and Schmidt, 1994; Reynolds *et al.*, 2000). Sometimes, it is also referred to as Generalized Likelihood Ratio (GLR) (Gish and Schmidt, 1994). LLR is used in this thesis in Chapters 4 and 5. In Chapter 5, we have used LLR to calculate a similarity measure between two PDFs.

Model Selection

In many situations, we are forced to choose among nested classes of parametric models, *e.g.* models with different number of parameters. The Maximum Likelihood (ML) principle (Edward, 1972) (*i.e.* maximizing the likelihood of available training data) was developed only for a single parametric family, and hence it is not guaranteed to yield a sensible selection criterion in such situations. Schwarz (1978) proposed a Bayesian approach to the model selection problem known as **Bayesian Information Criterion (BIC)**. BIC is an approximation to the posterior distribution on model classes. While based on the assumption that proper priors have been assigned to each model, this approximation effectively eliminates any explicit dependence on prior choice. The resulting solution takes the form of a penalized log likelihood. The BIC value for a model M is defined as:

$$BIC_M = \log p(X|M) - \frac{|M|}{2} \log N \tag{2.8}$$

where $p(X|M)$ denotes the maximum-log likelihood of data X given model M, $|M|$ denotes the number of free parameters in M and N denotes the number of observations in X. It was shown in (Schwarz, 1978) that maximizing BIC value also results in maximizing the integrated likelihood which is the expected value of the likelihood over the set of parameters of M. A model having maximum BIC value is selected using this theory. The consistency of BIC for GMMs was shown in (Biernacki *et al.*, 2000).

BIC was introduced for the case of speech and specifically for acoustic change detection and clustering by Chen and Gopalakrishnan (1998), where the problem was formulated as that of model selection. Since then, BIC has been used in many speech applications and is a state-of-the-art approach for acoustic change detection and clustering.

It is also interesting to note that BIC formally coincides with other information theoretic criteria like Minimum Description Length (MDL) (Rissanen, 1989) and Akaike Information Criterion (AIC) (Akaike, 1974). These information theoretic measures have a completely different motivation and derivation to BIC. The motivation for MDL, for example, is to select a model that provides the shortest description of the data, where describing data can be regarded as coding. The term depending on the number of free parameters in BIC (right hand side of (2.8)) is explained in the MDL framework as the extra cost incurred by transmitting the parameters of the model. Rissanen (1989) demonstrates that for a regular parametric family of dimension d, this amounts to transmitting at least $\frac{d}{2}\log N$ bits, where N is the length of the data.

2.2 Previous Works

This section presents a literature review of most of the significant work that addressed the issues related to audio segmentation and the three tasks addressed in this thesis namely, speech/music segmentation (Section 2.2.1), speaker change detection (Section 2.2.2) and speaker clustering (Section 2.2.3).

2.2.1 Speech/Music Segmentation

Like many other pattern classification problems, speech/music segmentation has two main modules; a signal processing based feature extraction module and a classification framework which uses these feature vectors. We now discuss some of the previous work related to these two modules for the purpose of speech/music segmentation.

Earlier work regarding the separation of speech and music classes addressed the problem of classifying known homogenous segments as speech or music. Earlier research also focused more on devising and evaluating characteristic features for classification. Saunders (1996) designed one such system to enable an automatic real-time radio channel surfing scheme for broadcast FM and investigated the use of zero-crossing rate (ZCR) to distinguish between speech and music signals. Sheirer and Slaney (1997) investigated the use of "low-energy" frames, spectral roll-off points (95^{th} percentile of the power spectral distribution), spectral centroid (correlate of ZCR) and spectral flux (delta spectral magnitude) and 4 Hz Modulation energy (syllabic rate of speech). K. El-Maleh and Kabal (2000) used line spectral frequencies (LSF) for speech/music discrimination. More recently, in a completely different approach, Williams and Ellis (1999) used posterior probability based features for this purpose.

The limitation of most of these works is that the systems were designed, and experiments were reported, only on discrete audio segments. These segments were meant to have sound of only one kind, i.e., either only speech or music. However, in most practical applications like broadcast news transcription, the audio stream is continuous, with speech and music interleaving randomly. Thus, deriving segmentation in terms of acoustically homogeneous segments along with classifying these segments is required in most practical applications and this is the problem that is addressed in this thesis.

Most recent work addresses this problem of segmenting a continuous audio stream, as part of a complete ASR system (Siegler *et al.*, 1997; Jain *et al.*, 1996; Spina and Zue, 1996; Moreno and Rifkin, 2000; Seck *et al.*, 1999). In most of this work, cepstral coefficients (*e.g.* MFCC) are used for segmenting and classifying speech and music signals, rather than using specially devised features such as those mentioned above. This is probably because MFCC coefficients are generally used for several modules in a big audio processing system, thus using them for speech/music segmentation avoids the need for calculating special features and saves computation. Moreover, although MFCC coefficients were originally designed to model the short-term spectral information of speech events for speech recognition, researchers have also studied their applicability for modeling music (Logan, 2000). These studies showed that the two major aspects of MFCC coefficients *i.e.* mel-frequency scale and DCT transform are well suited for modeling music signals as well. MFCCs have been used by several other researchers to model music and audio sounds (Foote, 1999; Blum *et al.*, 1999; Logan and Chu, 2000).

The use of specially devised features (Sheirer and Slaney, 1997; Saunders, 1996) or MFCC for speech/music segmentation has two limitations from an ASR point of view. The first limitation is that computing these features is an extra overhead because the features for phone recognition are computed separately. Secondly, the preprocessor for an ASR system should be such that it can filter out all the segments that can not be recognized, *e.g.* even filtering out all the Spanish segments from the input to an English recognizer. The specially devised features would not be able to do this because even these segments exhibit speech characteristics from a signal processing point of view. Moreover, in noisy situations when the signal-to-noise ratio drops, these specially devised features may classify these regions as non-speech although it may be possible for the ASR system to recognize these regions with state-of-the-art noise handling/cancellation techniques (Hermansky and Morgan, 1994). Moreover, although MFCC features are well suited for modeling music class (Logan, 2000), within the speech class these features are smoothed to hide and remove aspects of the signal that are not phonetically relevant, such as speaker identity and background noise, so we might expect these to form a poor basis for distinguishing speech from music signals.

2.2.2 Speaker Change Detection

Although speaker change detection also belongs to the family of pattern classification problems, and thus has a feature extraction module followed by classification/segmentation framework, no significant work has been reported on the feature extraction module. Traditionally, MFCC features are extracted every 10ms and fed to the segmentation algorithm.

However various segmentation algorithms have been proposed in the literature, which can be categorized as follows:

- Decoder-guided segmentation: The input stream can first be decoded; then the desired segments can be produced by cutting the input at the silence locations generated from the decoder (Kubala, 1997; Woodland *et al.*, 1997). Other information from the decoder, such as gender information could also be utilized in the segmentation.

- Model-based segmentation: This involves making different models e.g. GMMs, for a fixed set of acoustic classes, such as telephone speech, pure music, etc. from a training corpus (e.g. Bakis (1997); Kemp *et al.* (2000)); the incoming audio stream can be classified by ML selection over a sliding window; segmentation can be made at the locations where there is a change in the acoustic class.

- Metric-based segmentation: A distance-like metric is calculated between two neighboring windows placed at each sample; metrics such as Kullback-Liebler (KL) divergence (Siegler *et al.*, 1997), LLR (Bonastre *et al.*, 2000; Delacourt and Wellekens, 2000; Delacourt *et al.*, 1999; Mori and Nakagawa, 2002) or BIC (Chen and Gopalakrishnan, 1998; Delacourt and Wellekens, 2000; Delacourt *et al.*, 1999; Mori and Nakagawa, 2002; Lopez and Ellis, 2000; Zhou and Hansen, 2000; Tritschler and Gopinath, 1999; Vandecatseye and Martens, 2003) can be used. The local maxima or minima of these metrics are considered to be the change points.

All of these methods have limitations. The decoder guided segmentation only places boundaries at silence locations, which in general has no connection with the acoustic changes in the data. The model based segmentation approaches do not generalize to unseen data conditions as the models are no longer compatible with the new data conditions. The metric based approaches generally require a threshold/penalty term to make decisions. These thresholds are set empirically and require an additional development data.

Chen and Gopalakrishnan (1998) formulated the problem of speaker change detection as a model selection problem and used BIC (2.8) for this purpose. However, a tunable parameter was introduced in the penalty term. In practice, this parameter is used to improve the performance of the system for a particular condition. This parameter therefore implicitly defines a threshold. This was observed in our own experiments and also in several other previous works (Tritschler and Gopinath, 1999; Kemp *et al.*, 2000; Delacourt and Wellekens, 2000; Mori and Nakagawa, 2002; Vandecatseye and Martens, 2003; Lopez and Ellis, 2000). Later, Tritschler and Gopinath (1999) proposed several heuristics not only to make the algorithm of Chen and Gopalakrishnan (1998) faster, but also to give importance to detecting short changes (less than 2 seconds). There is also works which attempts to make the detection procedure faster by applying a distance measure prior to BIC. DISTBIC was another work in this direction by (Delacourt and Wellekens, 2000; Delacourt *et al.*, 1999).

Work	Summary
(Gish *et al.*, 1991)	Used LLR based Gish-distance for SCD
(Siegler *et al.*, 1997)	Used KL distance for SCD
(Chen and Gopalakrishnan, 1998)	Proposed BIC for SCD
(Tritschler and Gopinath, 1999)	Proposed several heuristics to speed up the BIC based SCD and to give importance to short speaker segments.
(Liu and Kubala, 1999)	Proposed phone based SCD using another penalized LLR
(Delacourt *et al.*, 1999)	Proposed using LLR prior to BIC to make it faster
(Kemp *et al.*, 2000)	Compared model based, metric based (KL, Gish-distance) and energy based approaches
(Bonastre *et al.*, 2000)	Used LLR for SCD
(Zhou and Hansen, 2000)	Proposed using T^2-statistics prior to BIC to make it faster
(Lopez and Ellis, 2000)	Used BIC based clustering to improve SCD
(Mori and Nakagawa, 2002)	Proposed VQ distortion measure for SCD and compared with BIC and LLR
(Vandecatseye and Martens, 2003)	Proposed a technique to normalize BIC

Table 2.1. Summary of previous work on speaker change detection (SCD)

An LLR based distance computation prior to BIC was proposed in this work, which is faster than BIC. Then, only selected change points were passed through the BIC test. In another work in this direction, Zhou and Hansen (2000) proposed applying T^2-statistics prior to BIC. Vandecatseye and Martens (2003) proposed a modification to BIC in terms of normalizing the BIC score and showed it to be better than the non-normalized BIC.

However, researchers have also explored techniques other than BIC and LLR for this task. Gish *et al.* (1991) proposed a distance measure for this purpose, which is referred to as Gish-distance in the literature. This distance is also based on LLR and was also used by Kemp *et al.* (2000). Mori and Nakagawa (2002) used a vector quantization (VQ) distortion criterion, while also comparing this to BIC and LLR. In an interesting study, Kemp *et al.* (2000) observed that model based approaches achieve high precision at moderate recall while metric based approaches are capable of achieving high recall at moderate precision. Thus, in (Kemp *et al.*, 2000), they combined the two approaches. Liu and Kubala (1999) introduced a new penalized LLR (BIC-like) criterion for speaker change detection. Another interesting point in Liu and Kubala (1999) was that the change points were detected based on the output of a speech recognizer. However, this approach also used thresholds, which were set to minimize the total error (false acceptance and false rejection).

Table 2.1 presents a summary of all the approaches and their comparison with other existing approaches.

2.2.3 Speaker Clustering

Most of the state-of-the-art solutions to this problem use a hierarchical clustering approach *i.e.* starting from a large number of segments (clusters), some of the clusters are merged following a "suitable" strategy. This strategy mostly consists of computing a distance metric between two clusters and then merging the clusters with the minimum distance. Since most of the popular distance matrices are monotonic functions of the number of clusters, an external method of controlling the number of clusters (or merging process) is also necessary and part of the problem. Thus, most of the previous work on this topic revolves around employing a suitable distance metric and corresponding stopping criterion.

One of the earliest pieces of work on speaker clustering from the point of view of speaker adaptation in ASR systems was proposed by Jin *et al.* (1997). In this work, the Gish-distance proposed in (Gish *et al.*, 1991) was used as a distance matrix, which is based on Gaussian models of the acoustic segments. Hierarchical clustering was performed on this distance matrix while selecting the best clustering solution automatically by minimizing the within-cluster dispersion with some penalty against too many clusters. The penalty term was needed because the within-cluster dispersion will keep monotonically decreasing leading to the unwanted clustering of one segment per cluster. Although, a study of the number of clusters obtained with different penalty terms was done, no systematic way was proposed here to deduce the optimal value of this term. It was also shown in this work that the automatic speaker clustering contributed significantly to reduction in the Word Error Rate (WER).

Siegler *et al.* (1997) used KL divergence (relative cross entropy) as the distance metric for this purpose. The KL distance between the distributions of two random variables A and B is an information theoretic measure equal to the additional bit rate accrued by encoding random variable B with a code that was designed for optimal encoding of A (Cover and Thomas, 1991). In an agglomerative clustering approach, the KL distance was also compared with the Mahalanobis distance. In this framework, an utterance was clustered with an existing cluster if it was within a threshold distance, otherwise it was used as the seed of a new cluster. Different threshold values were tried in this work but no systematic way of finding this threshold was proposed.

Solomonoff *et al.* (1998) used LLR and KL distance matrices for this purpose. An agglomerative dendogram clustering framework was proposed in this work. Thus, a closest pair of clusters was picked using these distance matrices and merged until there was only one cluster. In order to obtain the appropriate number of clusters, dendogram cutting was used. This was done using "cluster purity" as a measure, which was also proposed in this work.

A top-down split-and-merge clustering framework was proposed in Johnson and Woodland (1998); Johnson (1999). This work was based upon the idea that the output of this clustering was to be used for Maximum Likelihood Linear Regression (MLLR) adaptation, and so a natural evaluation metric for clustering is the increase in the data likelihood from adaptation. The distance matrices used for splitting and merging were Arithmetic Harmonic Sphericity (AHS) (Bimbot and Mathan, 1993) and Gaussian Divergence. The clustering scheme works top-down with each node being split into up to four child nodes at each stage. The splitting was done until no segments move or the maximum number of iterations was reached. At each stage of splitting, some of the nodes were merged. The merging criterion was based on simple distance from the center of the node to its segments. The decisions were made by comparing this distance against a threshold value. It was also found to be necessary to define when a split is allowable to prevent data being split back into its constituent segments. Thus, a heuristic based on minimum occupancy count was used to ensure robust speaker adaptation. This clustering algorithm was applied to the HTK broadcast news transcription system (Woodland *et al.*, 1997; Hain *et al.*, 1998).

The most commonly used distance metric for this purpose is BIC. BIC was proposed for speaker clustering by Chen and Gopalakrishnan (1998) who formulated the problem of speaker clustering as model selection. In this work, starting from each segment (hand segmented) as a cluster, hierarchical clustering was performed by calculating the BIC measure for every pair of clusters and merging the two clusters with the highest BIC measure (2.8). The clustering was stopped when no two clusters resulted into an increase in BIC measure, when merged. Although, in this work the authors used a theoretically motivated penalty value, subsequent work on speaker clustering using BIC (Tritschler and Gopinath, 1999; Lapidot, 2003; Vandecatseye and Martens, 2003) found that adjusting this penalty not only produces better results, but is also necessary for the system to run on unseen data.

Recently, in the framework of the DARPA-EARS program, NIST started the speaker diarization evaluation[1]. This task is very similar to speaker clustering, except that the motivation is a little

[1] http://www.nist.gov/speech/tests/rt/rt2003/index.htm

Work	Distance Matrix	Stopping/Merging criterion
Jin *et al.* (1997)	Gish-distance	Penalized within-cluster dispersion
Siegler *et al.* (1997)	Kullback Liebler distance	Thresholding KL distance
Solomonoff *et al.* (1998)	LLR and KL	Dendogram cutting using cluster purity
Johnson and Woodland (1998), Johnson (1999)	AHS	heuristics like minimum occupancy count threshold on Gaussian divergence
Chen and Gopalakrishnan (1998)	BIC	penalty to take care of model complexity
Tritschler and Gopinath (1999)	online BIC	penalty to take care of model complexity
Zhou and Hansen (2000)	KL	thresholding the KL distance
Lapidot (2003)	BIC-SOM	Used BIC-like penalty term in SOM
Vandecatseye and Martens (2003)	BIC	penalty against too many clusters
Gauvain and Barras (2003)	penalized log likelihood	penalty against too many clusters
Tranter *et al.* (2003)	AHS	same as Johnson and Woodland (1998)
Moraru *et al.* (2003)	BIC	penalty against too many clusters

Table 2.2. Summary of previous works done on speaker clustering

different. Speaker diarization is intended to make the transcription of ASR system richer by determining who said what? Whereas, the motivation of speaker clustering work, on the other hand, is to create speaker clusters from the point of view of speaker adaptation. Thus, in the later case, it is possible to make a single cluster for two speakers if the two individual clusters do not have enough data and the two speakers are acoustically very similar. Most of the approaches presented in this evaluation used BIC as the distance metric (Nguyen, 2003; Reynold *et al.*, 2003; Moraru *et al.*, 2003). Another penalized LLR distance matrix was proposed by Gauvain and Barras (2003), where the penalty parameters were tuned to get the best performance. The Cambridge diarization system (Tranter *et al.*, 2003) used the framework defined in (Johnson and Woodland, 1998).

Table 2.2 summarizes these works from the point of view of the distance metrics they used and how the stopping criterion was formulated. Thus, it is clear from Table 2.2 that BIC is the state-of-the-art approach toward speaker clustering. More discussion about BIC will follow in Chapter 4, however, the important point to note about this approach (in the framework of speech) is the use of a hidden threshold in the form of a penalty factor. A comprehensive study of the effect of this penalty factor on the segmentation accuracy for the broadcast news transcription task is presented in (Tritschler, 1998).

Chapter 3

Speech/Music Segmentation

3.1 Introduction

Speech/music segmentation is the task of partitioning an audio stream into speech (somebody speaking) and music segments. This is the first fundamental step toward content based audio information retrieval because most of the signal processing algorithms are separately designed for these classes. A good segmentation in terms of speech and music has many practical applications. For example, traditionally, separate codec designs are used to digitally encode speech and music signals to ensure a low bit-rate. An effective speech/music segmentation would enable these to be merged into a universal coding scheme. It could prove to be very helpful in automatic navigation of audio documents or automatic switching of TV channels. From an ASR system point of view, this problem is even more useful because the recognizers are only trained to recognize speech sounds. In audio recordings like broadcast news speech, speech is often interleaved with non-speech (predominantly music) segments. When indexing or summarizing these recordings, filtering out all the segments that cannot be recognized from the input audio stream will not only result in reduced computational complexity but also help in generating more understandable and accurate output.

Several works have been reported in the literature as already discussed in Chapter 2. Contrary to previous approaches, where the speech/music segmentation was based on the acoustic nature of the signal, the framework proposed in this chapter is primarily based on the functioning of an ASR system itself. In other words, the proposed system is designed to discriminate all recognizable speech segments from all others which cannot be recognized. This is very useful as pre-processing for an ASR system, as it results in lower computational complexity and more accurate output. This also has the obvious advantage over previous approaches that it does not simply look at the acoustic nature of the signal to classify it as speech or music but looks at how well the recognizer can perform over these segments. To further clarify this, let us take the example of a Spanish speech segment fed as an input to an English recognizer. The previous acoustic based approaches would tend to classify this as a speech segment, but from an ASR point of view, it is still not recognizable and should be discarded. On the other hand, let us consider another example where the speech is accompanied with a very high level of background noise resulting in a very low signal to noise ratio (SNR). Previous acoustic based approaches (especially if they are trained on clean speech and music signals) might tend to classify this segment as non-speech. It is possible, however, that the recognizer is able to take care of this background noise by applying appropriate noise handling/canceling techniques (*e.g.* Hermansky and Morgan (1994)) and therefore should not be prevented from being recognized.

The feature vectors and framework explained in this chapter address these drawbacks as they are directly based on the functioning of an ASR system. We have chosen a HMM/ANN hybrid ASR system (Bourlard and Morgan, 1994) for this purpose. In this framework, the emission probabilities of different states of the HMM are estimated via an MLP. This MLP is trained on clean speech and estimates the posterior probabilities of the phonemes used for recognition. If we employ an information theoretic measure, namely entropy (Papoulis, 1991), at the output of this MLP (which can be seen as a channel in this context), it is expected that the entropy (basically a degree of disorder) at the output of this channel should be low when the input is a clean speech signal and high when the input is non-speech. On the other hand, these probabilities are expected to be changing faster (with changing phonemes) during speech regions exhibiting higher "dynamism" (mathematically defined later) and slower during non-speech (music in this context) regions. The work presented is basically motivated by this expected phenomenon.

The two features, entropy and dynamism, were introduced for the purpose of speech/music discrimination by Williams and Ellis (1999). The basic motivation for extracting these feature vectors is to carefully analyze the behavior of the posterior probabilities. We note that there are various other ways of computing feature vectors from these probabilities such as phone distribution match and background-label energy ratio (also proposed in (Williams and Ellis, 1999)), or computing Kullback-Liebler (KL) divergence between two successive frames. The work presented in this chapter builds upon the work presented in (Williams and Ellis, 1999) by these two features in a HMM framework. Two different approaches for estimating the emission probabilities of states of this HMM are investigated, namely GMM and MLP experts. Moreover, a detailed analysis of the relative behavior of entropy and dynamism features for speech and music classes and also their comparison with MFCC acoustic vectors is carried out in this chapter. The robustness of the proposed approach is demonstrated with different experiments, including different speech and music styles, as well as different temporal distributions of the speech and music signals (real data distribution, mostly speech or mostly music).

However, in many practical situations such as broadcast news speech, speech is accompanied by music or other noisy signals. These segments are referred to as "mixtures" in this thesis. As is the case for human beings, it may or may not be possible for an ASR system to recognize the signal in these situations depending on the proportion of speech in the signal. This chapter also presents a method to compute a confidence measure which basically indicates the percentage of recognizable speech in the signal. With the help of ASR experiments, it is shown that this confidence score is a good indication of the resulting WER *i.e.* segments with high (low) speech confidence consistently result in low (high) WER. Moreover, since broadcast speech not only has music as a non-speech category but also other kind of noisy sounds (*e.g.* clapping, background noise, channel noise during tele-conferencing etc.), these ASR experiments also show the effectiveness of the proposed approach for detecting non-speech signals in general. In summary, given an ASR system, the system in this chapter is shown to discriminate all "recognizable" segments from all the other "non-recognizable" segments, while also providing a confidence score of "recognizability" of that segment.

This chapter is organized as follows: First, the mathematical definitions and interpretations of the posterior probability based features, namely entropy and dynamism, is presented in Section 3.2. The HMM framework in which these features are used is discussed in Section 3.3. Section 3.4 then presents the implementation details, evaluation set-up and the performance of the system on different test datasets. The confidence measure used to handle "mixture" situations is presented in Section 3.5.1 and an analysis of how this is an indication of the resulting ASR performance is also presented in this section. Finally, a conclusive summary of this chapter is presented in Section 3.6.

3.2 Posterior probability based features

According to information theory, a channel designed for a particular type of signal will exhibit characteristic behavior at its output when that signal is passed through the channel. Conversely, the presence of a different type of signal will result in uncharacteristic behavior at the channel output. In the case where the channel is an MLP trained to emit posterior probabilities for speech recognition as in hybrid ASR, it should therefore be possible to distinguish between speech and non-speech signals by examining the behavior of these probabilities. In this section, and building upon (Williams and Ellis, 1999), we investigate two features, namely *entropy* and *dynamism*, with which we can analyze the behavior of these posterior probabilities.

3.2.1 Entropy

Entropy is a measure of the uncertainty or disorder in a given distribution (Papoulis, 1991). In the case of an MLP trained to emit posterior probabilities for K output classes (usually associated with speech phones or HMM states q_k, $k = 1, \ldots, K$), the *instantaneous entropy* h_n at a specific time frame n is defined as:

$$h_n = -\sum_{k=1}^{K} P(q_k|x_n) \log_2 P(q_k|x_n) \tag{3.1}$$

where x_n represents the acoustic vector (PLPC) at time n, q_k the k-th MLP output class, and $P(q_k|x_n)$ the posterior probability of class (phone) q_k given x_n at the input.

The posterior probabilities at a given time and given current observation vector represent a true probability distribution, and the entropy of this distribution (the expected value of the log probability) is a measure of the goodness-of-fit of the current observation to the acoustic model (channel). Generally, in the case of speech, the value of the posterior probability for a particular phoneme (the 'recognized' phoneme) is much higher than other phonemes. This means that the value of the entropy will be close to zero, indicating that little information will be gained by knowing its actual value, or, equivalently, that there is little uncertainty over the unknown segments. In the case when a music signal is passed through the MLP, the values of the probabilities will be more uniformly distributed, resulting in a higher value for entropy. Eq. (3.1) gives the instantaneous value of the entropy at frame n. As we will see in the subsequent discussion, it is advantageous to average this instantaneous entropy over a window of several frames, resulting in the *averaged entropy* at time n:

$$H_n = \frac{1}{N} \sum_{t=n-N/2+1}^{n+N/2} h_t \tag{3.2}$$

where n is the index of the current acoustic frame and N is the size of the averaging window.

3.2.2 Dynamism

Dynamism is a measure of the rate of change of a quantity and was introduced by Williams and Ellis (1999). In this case, and using the same notation as above, the *instantaneous dynamism* at time n is defined as:

$$d_n = \sum_{k=1}^{K} [P(q_k|x_n) - P(q_k|x_{n+1})]^2 \tag{3.3}$$

This feature captures the dynamic behavior of the probability values. As speech is a highly varying signal, it can be expected that the probability values observed at the output of the MLP in the case of speech input will change rapidly from one frame to another. Conversely, music is a harmonic (less varying) signal and hence the dynamism is low, resulting in higher dynamism for speech than for music.

Similar to the case of entropy, it can be beneficial to average the instantaneous values of dynamism over a certain number of frames, resulting in the *average dynamism* at time n:

$$D_n = \frac{1}{N} \sum_{t=n-N/2+1}^{n+N/2} d_t \tag{3.4}$$

where N is the size of the averaging window.

3.3 Speech/Music Segmentation

The complete block diagram of the proposed speech/music discrimination system is shown in Figure 3.1. In this figure, the primary[1] MLP refers to the one used in standard hybrid ASR. In practice, this MLP estimates the posterior probabilities of the phonemes given feature vectors corresponding to a temporal contextual window of a certain duration (typically 9 acoustic frames of 16-ms), i.e., $P(q_k|x_n)$ where q_k is the phonetic class (with $k = 1, \ldots, K$, where K is the total number of output classes) and x_n is the feature vector at time n.

Average entropy H_n and Average dynamism D_n features are extracted from these posterior probabilities according to (3.2) and (3.4), respectively, at every time n. These features form a two-dimensional vector $y_n = (H_n, D_n)^T$, which is then used as the HMM observation vector. The individual PDFs (modeled as a GMM) of entropy and dynamism are shown in Figures 3.2 and 3.3, respectively. It is clear from these figures that the behavior of these features for speech and music is quite distinct and that they can be effective discriminatory features. The figures also show both the distributions for the instantaneous (local) values of entropy h_n (3.1) and dynamism d_n (3.3), as well as the average entropy H_n (3.2) and average dynamism D_n (3.4). From the figures it is clear that the averaged measures exhibit better Gaussian properties (due to the central limit theorem), and are more clearly separated and thus better suited for discrimination purposes.

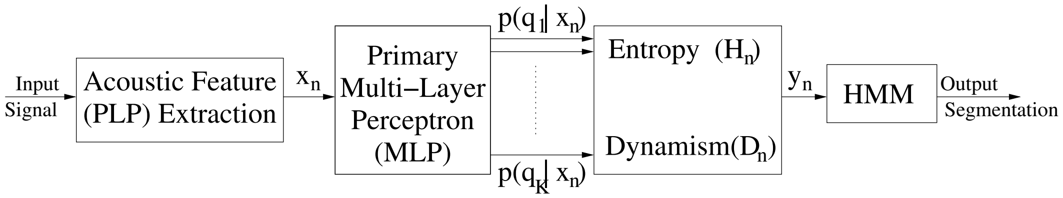

Figure 3.1. Block diagram of the proposed speech/music segmentation system. PLP Cepstral coefficients (x_n) are extracted every 16ms. These coefficients are then fed to the input of an (primary) MLP trained on clean speech for ASR. The output of this MLP, the posterior probabilities, are then analyzed in the form of entropy (3.2) and dynamism (3.4). The two-dimensional vector, y_n composed of entropy and dynamism is input to a HMM classifier. The emission probabilities of the HMM are estimated using a GMM or a secondary MLP expert.

The PDFs of the two classes as shown in Figures 3.2 and 3.3 are modeled by GMMs with 5 mixtures. Although these distributions may appear Gaussian, especially with the averaging, it was observed in our experiments that a single Gaussian density could not model the full variability and resulted in inferior performance. On the other hand, the exact value of the number of mixtures

[1]We refer to this MLP as primary to avoid the ambiguity with another MLP used at a later stage in this work

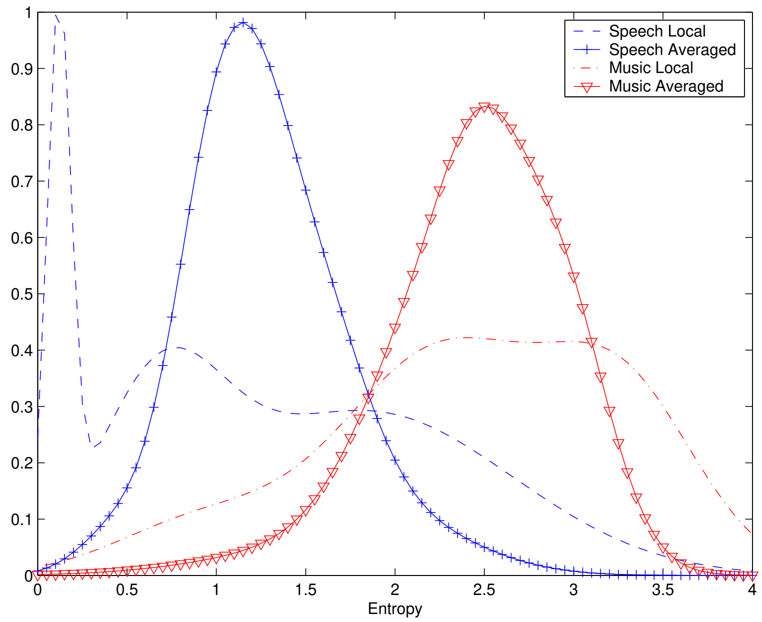

Figure 3.2. Distribution of local and average entropy for speech and music. As expected, the average entropy is usually higher for music than for speech.

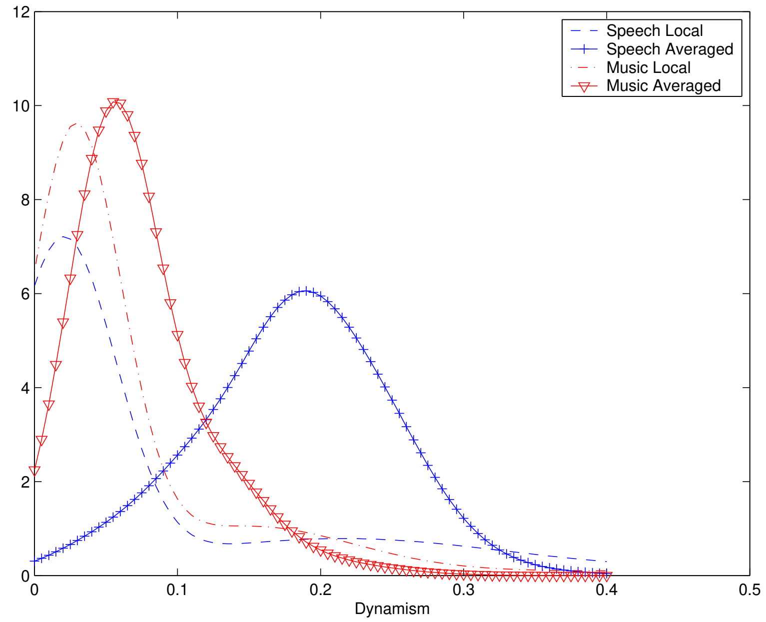

Figure 3.3. Distribution of local and average dynamism for speech and music. As expected, the average dynamism is usually higher for speech than for music. It should be noted that the histogram shows values only from 0 to 0.4 on the dynamism-axis to highlight the difference between speech and music distributions and also to highlight the discriminatory advantages of averaged dynamism over local dynamism.

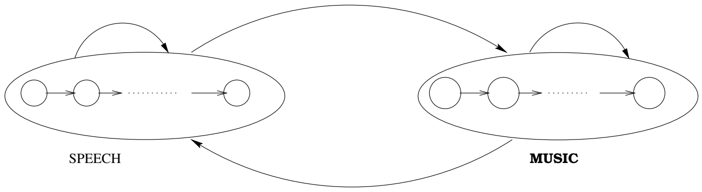

Figure 3.4. HMM topology for the proposed speech/music segmentation system. The two classes are represented by hyper-states composed of several sub-states to impose minimum duration constraint. The emission probabilities of these sub-states of the same class are estimated by same GMM or MLP expert.

(greater than 1) did not significantly affect performance. These GMMs are trained via the EM algorithm in a supervised way. We also investigated the possibility of training these parameters of GMMs in an unsupervised way using exactly the same HMM topology as shown in Fig. 3.4 (Ajmera *et al.*, 2002). In our experiments (shown later in Section 3.4.3), we found that the PDFs trained in an unsupervised way provide similar performance as compared to the PDFs trained in a supervised way. This is very useful because it not only alleviates the need for labeled training data but also shows that the system can be adapted to non-speech sounds other than music.

We also investigated the use of an MLP expert (different from the primary MLP) for this purpose. The motivation for this was to investigate and exploit the discriminatory features of MLP training. This MLP was trained via the EBP algorithm on labeled training vector sequence $y_n = (H_n, D_n)^T$. At the time of segmentation, given a vector of entropy and dynamism values, this MLP outputs the posterior probabilities of speech and music classes, which can be turned into scaled likelihoods to be used in the HMM, following (2.5). Also, since an equal amount of speech and music data was used for training this MLP, the prior probabilities were equal and could be dropped. Approximately 2.5 hours of audio data was used for training the GMM and MLP experts. Specifically, we used speech data from BBC news broadcast database recorded between 1996 and 1998, and music files from the THISL music database which contains different types of music (rock, pop, classical etc.) recorded from different Swiss radio stations in 1997.

3.3.1 HMM Classifier

The HMM topology for the proposed system is shown in Figure 3.4. This HMM is a 2-state fully connected model, where a minimum duration is imposed for each state. This is achieved by simply concatenating internal states associated with the same PDF.

As there is no *a priori* information available regarding transitions across speech and music segments, transition probabilities are set manually to favor remaining in the current (speech or music) state. Similarly, as there is no information regarding the beginning of the audio data, initial probabilities are set manually to make speech and music segments equally likely. The emission probabilities for the HMM states ($p(y_n|q_k)$) are estimated by either a GMM (2.3) or an MLP (2.5) expert. The Viterbi algorithm is then used to find the best possible state (speech/music) sequence which could have emitted this observation sequence.

3.4 Evaluation Experiments

3.4.1 Implementation

For the posterior probability calculation, we use a (9x13)-2000-42 MLP[2]. The input acoustic features are the first 13 PLPC coefficients extracted every 16 ms.

For the purpose of entropy and dynamism calculation, the number of phonemes K is 42 (number of output units of the MLP) and the size of averaging window N is 40. Different averaging windows were tried for this purpose on a development dataset and $N = 40$ was found to be the optimal window size for this purpose. Also, we expect at least a couple of phoneme-transitions within this window to differentiate between the properties of speech and music classes, which is specially true for the case of dynamism.

The number of states used to impose the minimum duration constraint in the HMM was fixed to 180, thus assuming in our case that any speech or music segment is never shorter than 2.88 seconds (16ms $\times$ 180). The self-loop probabilities were set to 0.9 for the last state of each class. In our experiments, we observed that this self-loop transition probability did not affect the system performance as long as values greater than 0.5 (favoring self transitions) were used.

To assess the effectiveness of the entropy and dynamism features, a baseline system using 24-dimensional MFCC coefficients (without delta and double-delta terms) in a HMM/GMM framework (using the same topology as in Figure 3.4), was also included in the evaluation for comparison purposes.

[2]This MLP was trained as a part of the THISL Project (1997)

3.4.2 Evaluation

We evaluated the system using 4 labeled data sets, each 10 minutes long. These data sets have a wide variety of speech and music. For example, they contain speech from a variety of both male and female speakers, as well as different types of music, such as jazz, pop, and country. To observe the effect of durations of sounds, segments of different durations have been mixed together. Three different experiments were performed on each of these data sets using both GMM and MLP experts.

Results were obtained in terms of the percentage frame level accuracy. We calculate three different statistics in each case : the percentage of true speech frames identified as speech, the percentage of true music frames identified as music, and the overall percentage of speech and music frames identified correctly.

3.4.3 Results

Test Set 1

This is a 10 minute audio stream having alternate speech and music segments of equal (15 seconds) duration. The classification results are shown in Table 3.1.

Expert	*Feature*	*Speech*	*Music*	*Total*
GMM	Entropy	98.7	92.7	95.6
GMM	Dynamism	98.9	64.9	81.8
GMM	Both	98.8	93.9	96.2
MLP	Entropy	98.8	93.7	96.2
MLP	Dynamism	95.0	74.8	84.8
MLP	Both	97.4	93.7	95.5
GMM	MFCC	94.3	91.6	92.8

Table 3.1. Speech/Music classification results for Test Set 1

In this case, both entropy and dynamism features are capable of identifying the speech segments with a high degree of accuracy, as is required in ASR applications. However, using dynamism only, the frame level accuracy for music segments is very low. We found that some of these segments contained rap music, with less instrumental music in the background. Still, entropy performs well in discriminating these segments from speech. It is also clear that the performance of these two features is better than that of the 24-dimensional MFCC features. The performance of the GMM and MLP experts are comparable in this case.

Test Set 2

The second test set, which presents a more realistic task, consists of a 10 minute audio stream containing varying lengths of alternate speech and music segments. These segments include very short (2 seconds) as well as long (14 seconds) segment durations. These classification results are shown in Table 3.2.

Expert	*Feature*	*Speech*	*Music*	*Total*
GMM	Entropy	91.4	97.5	94.2
GMM	Dynamism	99.6	79.4	89.0
GMM	Both	98.1	91.6	94.5
MLP	Entropy	97.4	92.4	94.6
MLP	Dynamism	93.0	84.8	88.6
MLP	Both	98.6	94.6	96.3
GMM	MFCC	93.9	92.2	92.9

Table 3.2. Speech/Music Classification results for Test Set 2

The performance of different features and experts for this data set follows the same trend as in test set 1. This indicates the robustness of the proposed system to sound durations. To test the significance of the minimum duration constraint, some segments shorter than minimum duration were included in this test set, with the expectation that they should be filtered out. Examination of the output transcripts shows that these undesired short segments are correctly ignored in the case of the entropy and dynamism features. In contrast, due to the slower and less abrupt variations in the features, these segments appear in the MFCC system, borrowing frames from the neighboring segments to respect the minimum duration.

Test Set 3

The third test set consists of a 10 minute audio stream comprised mainly of speech data. In this case, 15 second segments of speech data are interleaved with short segments of music. This represents a more likely scenario for the case where the speech/music discrimination is being used as a pre-processing step to segment a predominantly speech audio signal for automatic speech recognition. These classification results are shown in Table 3.3.

Expert	*Feature*	*Speech*	*Music*	*Total*
GMM	Entropy	96.9	82.8	91.5
GMM	Dynamism	91.7	82.8	88.3
GMM	Both	97.0	93.6	95.6
MLP	Entropy	94.8	87.2	91.8
MLP	Dynamism	83.5	91.0	86.2
MLP	Both	91.6	96.4	93.2
GMM	MFCC	95.3	87.8	92.4

Table 3.3. Speech/Music classification results for Test Set 3

The advantage of combining the two features becomes evident from the results of this test set, clearly improving the total performance and the performance over music segments.

Test Set 4

The final test set contains a 10 minute audio stream mostly consisting of music data. In this case, 15 second music segments are interleaved with short segments of speech. The classification results are shown in Table 3.4.

Expert	*Feature*	*Speech*	*Music*	*Total*
GMM	Entropy	93.3	91.1	91.8
GMM	Dynamism	98.3	70.8	81.0
GMM	Both	97.2	92.7	94.3
MLP	Entropy	93.3	91.0	91.7
MLP	Dynamism	90.7	83.7	86.2
MLP	Both	91.6	96.4	93.2
GMM	MFCC	95.9	92.2	93.5

Table 3.4. Speech/Music classification results for Test Set 4

We note that the speech segments are detected with a high degree of accuracy, despite having been interleaved with large segments of music. This ensures no loss of information when speech/music discrimination is carried out as a preprocessing stage for speech recognition.

Unsupervised Training

As mentioned in Section 3.3, we also explored the possibility of training the PDFs of the two classes in a completely unsupervised way. This was done using the Viterbi approximation of the EM training of HMMs as explained in Chapter 2, using the HMM topology defined in Section 3.3.1 and as shown in Figure 3.4. The parameters of the GMMs were initialized using K-Means algorithm, while also utilizing the *a* priori information that the entropy (dynamism) values are lower (higher) for speech than that of music. The other parameters of HMM (transition and initial probabilities) were set manually based on the experiences from supervised training mentioned in previous section. and Table 3.5 shows the comparative results of the systems when the PDFs were trained in a supervised and unsupervised way. The results in this table show that both the systems perform very similarly on average.

Test Set	Supervised	Unsupervised
1	96.2	95.7
2	94.5	94.8
3	95.6	95.0
4	94.3	94.9
Average	94.2	94.1

Table 3.5. Comparative study of speech/music segmentation systems using supervised and unsupervised training methodologies. The results show that both the systems perform very similarly on average.

3.4.4 Discussion

The observations from these four data sets can be summarized as follows:

- The two dimensional entropy and dynamism feature vectors show statistically significantly better performance (overall 95.2%) in discriminating speech and music classes compared to standard 24-dimensional MFCC features (92.9%)[3]. This can also be attributed to the more training went in the proposed framework as the primary MLP is trained with many more examples of clean speech.

- Overall, entropy is a better discriminatory feature than dynamism, especially during music segments. Both features are individually capable of detecting speech frames with a high degree of accuracy. The combination of the two features, in some cases, significantly improves the results, showing that the two features have complementary information. This can be attributed to the fact that entropy captures the frame-level behavior whereas dynamism captures the temporal behavior, of the posterior probabilities.

[3]As shown by standard proportion test (commonly known as z-test) with 99% confidence

- Dynamism fails to detect music frames with a high degree of accuracy, especially if the music is composed of more vocal sounds than instrumental music, as in the case of rap music. However, entropy still performs adequately in this situation. This shows that the output of the primary MLP is still music-like in nature (high entropy, probabilities uniformly distributed) within a frame, but changes rapidly between frames, giving higher values of dynamism.

- In general, the relative behavior of entropy and dynamism does not change in the GMM and MLP frameworks. The performance of the two experts (GMM/MLP) is also comparable in all cases.

- In the framework of speech recognition, an important advantage of audio segmentation is the saving in computation time for non-speech segments. When the proposed system is used in conjunction with a hybrid HMM/MLP speech recognition system, computation is reduced, as music segments are not passed to the Viterbi decoder, which is the most computationally intensive element of the hybrid recognizer.

- The fact that unsupervised training of the parameters of the GMMs provides similar performance to the supervised training is useful for two reasons: First, this eliminates the need for labeled training data. Second, it shows that the system can easily be adapted to more general speech/non-speech classification problems.

As an aside, we note that some of the error may be attributed to the inherent latency of the system. At a first level, a high amount of averaging is done in the pre-processing stages in order to extract the speech recognition features and contextual information for input to the MLP. This is followed by another level of averaging to obtain the average entropy and average dynamism features. Due to these factors, the features will not change abruptly as the signal makes a transition from speech to music and vice-versa. Another level of latency is introduced by the minimum duration constraint within the HMM, where a specified minimum amount of time is required to decide whether the signal is really a music or a speech signal. The combined effect of these factors will mean that perfect 100% accuracy at the frame level is unlikely to be achieved in practice.

3.5 Extension and Application to ASR

From the above, it is clear that the framework presented in this chapter is based upon the functioning of a hybrid ASR system. It is also clear that such a system would help ASR by discarding all "non-recognizable" segments and thus saving computation time, while also leading to a more understandable and accurate output. However, in many practical situations like broadcast news speech, the classification of the data in terms of speech and non-speech (or music) is not straightforward even for a human-being. This is because, speech is often accompanied (overlapped) with other sounds including music, background noise, etc and these regions are referred to as "mixtures". The proportion of speech in the signal is constantly varying during and across these "mixture" regions. In fact even during clean speech regions (clean for a human listener), the SNR is not perfect (infinity). Thus, an automatic way to compute a measure of the proportion of speech in the signal would be very useful for ASR. For example, depending on the application, all the regions below a certain proportion of speech can be dropped from recognition. Moreover, this quantity is also like a meta-data information over the ASR output. In the following, we define a confidence score based on our speech/music segmentation system which can serve the above mentioned purpose of indicating proportion of speech and also investigate its usefulness for ASR.

3.5.1 Confidence Measure

In the context of the MLP system (system which uses MLP expert to estimate emission probabilities of the states of HMM), we obtain the posterior probabilities for the speech and music classes for each input feature vector y_n. For a segment of length N we can define a measure of the confidence of the speech class (as an indication of the proportion of speech in that segment) from the arithmetic mean of these frame probabilities. We adopt the arithmetic mean in this case so that the segmental confidence measure is not unduly biased by the probability estimates of a single frame (it is evident that use of the geometric mean would result in an average confidence of 0 if only one of the frames gave a probability of 0). Thus, the confidence score for a segment of length N is defined as:

$$Conf(Speech) = \frac{1}{N}\sum_{n=1}^{N} P(Speech|y_n) \qquad (3.5)$$

This confidence score has a range $0 \leq Conf(Speech) \leq 1$ and obeys the constraint that $Conf(Speech)+ Conf(Music) = 1$.

Such a measure would have application, for example, in the context of a multi-modal fusion application in which the speech/music segmentation information, and indeed the speech recognition output, are simply input cues (or features) for higher-level processing decisions combining cues from different modalities. Such a technique has been applied in the framework of the European project ASSAVID (ASSAVID Project, 2000), which is concerned with automatic indexing of sports videos, and a demonstration of initial results of the scheme on a sample audio segment is available at `http://www.idiap.ch/~jitendra/speech-music`. The demonstration consists of an MPEG file which plots the value of the speech confidence measure calculated over segments as the audio signal is played.

We observed that segments with very high confidence values are usually clean and easily recognizable speech. It is important to emphasize the "recognizable" part because we noticed that even a clean speech segment but with a different accent of the same language (*e.g.* American English accent as an input to our system which is based on an ASR system trained on British English accent) may result in a lower confidence score. On the other hand, segments with very low confidence score are mostly music (*e.g.* music immediately before and after a show or advertisements during commercial breaks) or other environmental noises. Speech in a different language also falls in this category. The regions between these two situations are classified as "mixtures". These regions are speech accompanied with music or other environmental sounds. Sometimes, clean speech over a different channel (*e.g.* a conversation via teleconferencing) or in a different accent (but in the same language) also has a moderate confidence score. This analysis is further verified with the help of ASR experiments reported in the following section, where we see that this confidence score indeed is an indication of resulting ASR performance in terms of WER. The concept of identifying mixtures along with speech and music classes has also been used in Chapter 6.

3.5.2 Relation to ASR Recognition Accuracy

The purpose of the following study is to investigate the correlation between the confidence score computed using (3.5) for each segment generated by the speech/music segmentation system and the performance of an ASR system in terms of Word Error Rate (WER) for these segments. Two

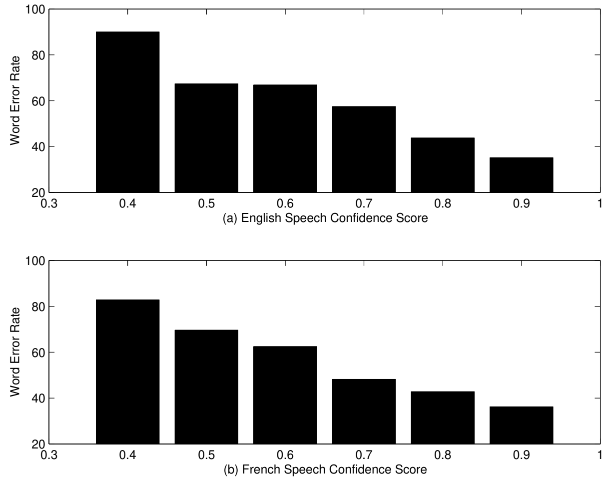

Figure 3.5. Word error rate as a function of confidence score for (a) English and (b) French systems. Values given are word error rates for all segments in the different confidence score ranges (i.e. $x > 0.9$, $0.8 < x < 0.9$, ..., $0.4 < x < 0.5$).

different speech/music segmentation systems based on two ASR systems (English and French) were investigated for this purpose.

The test data used for the English system this analysis was one hour of BBC broadcast news speech for English system. This data included clean speech, noisy speech and music/non-speech segments. This English ASR system is the same as explained in Section 3.4.

The French speech recognition system was based on the system presented in Andersen (1998), which achieved a best performance of 23.3% word error rate on the BREF read-speech corpus. The primary MLP for the system was re-trained on the CIMWOS broadcast news corpus. The speech in this corpus showed a high variability in acoustic conditions, ranging from clean speech from the in-studio news presentation to low quality speech in considerable background noise from field reporters. The test data for the French system was 2 hour French television news speech.

Figure 3.5 clearly shows the correlation between the proposed confidence score and the recognition accuracy. It is also evident from the figure that the proposed strategy for both English and French ASR systems basically resulted in segmenting the data into "recognizable" and "non-recognizable" segments and thus, segments with very low confidence do not need to be sent to the recognizer.

3.6 Conclusion

The goal of this chapter was to introduce a new approach for speech/music segmentation. This approach is primarily suited for an ASR framework where the goal of any such pre-processor should be to discriminate recognizable from non-recognizable segments. In the proposed system, this is achieved by using more appropriate features, which are basically based on the performance of the recognizer itself and independent of the acoustic properties of the signal.

For this purpose, *entropy* and *dynamism* features based on posterior probabilities of speech phonetic classes are used to form a two-dimensional observation vector sequence which is used in a HMM classification framework. A detailed study of these features and their relative performance to discriminate between two sounds are presented in this chapter. We have also studied and compared the use of both GMM and MLP experts to estimate the emission probability distribution of the HMM states. The relative performances of entropy/dynamism and GMM/MLP are demonstrated and discussed in the context of an experimental evaluation. Moreover, the performance of these two features is also compared with MFCC.

The system was tested with different speech and music styles, as well as different distributions of speech and music signals. The results of these tests illustrate the robustness of the approach, with the system achieving consistent frame accuracies from 93% to 96% across a variety of realistic test scenarios. From these results, we conclude that entropy and dynamism together make a powerful feature set for speech/music discrimination.

It was shown that the two-dimensional feature vector shows better performance compared to standard 24-dimensional MFCC features. While analyzing the relative performance of the two features, we found that overall, entropy is a better feature than dynamism for detecting music segments, while both features perform very well for detecting speech segments. Moreover, the fact that the combination of the two features always results in better discrimination accuracy, indicates the complementary information carried by the two features.

The overall performances of the GMM and MLP experts are comparable in all the cases. It was shown that the performance of the system based on PDFs trained in unsupervised way is very similar to that of unsupervised PDFs. This feature can be used to adapt the system to various non-speech categories in unseen conditions.

On the other hand, the MLP expert based system holds an advantage in terms of leading to a confidence measure as a simple and meaningful (values ranging between 0 and 1) indication of the proportion of the speech in a given audio segment. Such a confidence measure was defined and an analysis of the variation in this confidence score with the varying nature of the input audio signal was presented. The potential use of such a confidence measure in the context of speech/music mixtures was also briefly discussed. Furthermore, with the help of ASR experiments on two systems (English and French), it was shown that this confidence score is also an indication of the resulting ASR accuracy *i.e.*segments with high speech confidence are likely to have higher recognition accuracy as well and vice-versa. Thus, depending on the task, segments having very low confidence score can be prevented from being recognized thereby saving computation time as well as resulting in more accurate and understandable output.

In summary, the proposed speech/music discrimination system provides a powerful and robust pre-processing technique for reliable segmentation of audio streams in terms of recognizable and non-recognizable speech for an ASR system.

Chapter 4

Speaker Change Detection

4.1 Introduction

Speaker Change Detection (SCD) is the problem of finding speaker segment boundaries *i.e.* time instants when a speaker begins and stops speaking in a given audio stream. It is also considered a part of the problem that no information about the sought speakers and their number is known *a priori*.

This segmentation of audio data is of interest to a broad class of applications, like surveillance, meeting summarization or indexing of broadcast news. There are several ways in which SCD could prove to be useful for these applications. First, ASR technology requires segments of relatively short length as input and a SCD step provide one such segmentation. Second, as speakers tend to repeat within an audio stream, speaker adaptation schemes can be used to improve ASR performance. This is usually done by one such SCD turn, followed by a clustering step. These clusters are then used as a basis for speaker adaptation. Thus, a segmentation that is used for speaker adaptation needs to have a high *purity i.e.* one segment should contain only one single speaker and one acoustic condition (which is essentially the goal of SCD). In an audio indexing system, SCD provides a structural summary of the speaker turns contained in a conversation and could also prove to give an effective cue for boundaries between programs, topics, or scenes in a multi-media application.

Several previous approaches toward SCD have already been discussed in Chapter 2. It was mentioned that these approaches can be broadly categorized as decoder based, model based and metric based approaches. The most common approaches used for SCD are metric based, mostly because these approaches have the advantage that they do not rely on any pre-existing models, which tend to fail in unseen conditions. In fact, the models in these approaches are built and updated on the fly as new audio data arrives. A systematic formulation of the SCD problem in the framework of metric based approaches will be presented in Section 4.2. However, as pointed out in Chapter 2, most of the previous metric based approaches still rely on threshold or penalty-like terms, making them less robust to unseen data conditions. A mathematical description of this limitation for popular distance metrics will follow in Section 4.3.

This chapter proposes a novel metric based approach which overcomes this limitation by employing a distance metric which can be used to make decision about a potential change point without using any adjustable threshold or penalty term, therefore, eliminating the need for any development data to adjust this value. This is basically achieved by making the distance metric a function of likelihood ratio of two models with the same complexity. A mathematical description and an intuitive explanation of this will be presented in Section 4.4. Section 4.5 then presents experiments conducted on Hub4 1996 evaluation dataset to prove the robustness of the proposed approach and to compare the performance of proposed method with most commonly used approach, namely the BIC.

4.2 Formulation of the problem

Most of the metric based approaches toward speaker change detection formulate the problem like this: to decide if a speaker change point exists at time t or not, two neighboring windows of relatively small size are considered as shown in Figure 4.1. The content of these windows are usually sequences of acoustic feature vectors extracted from the audio signal. In Figure 4.1, these sequences are denoted as $X = \{x_1, x_2, ..., x_{N_x}\}$ and $Y = \{y_1, y_2, ..., y_{N_y}\}$, where N_x and N_y are the number of data points in the two windows respectively. Let Z denote the union of the contents of the two windows having $N = N_x + N_y$ data points. The contents of these two windows are compared using a dissimilarity function. Local optima of this dissimilarity function, compared to a threshold, are considered to be speaker change points.

It is also a common practice to model the datasets X, Y and Z as Gaussian PDFs. The ML esti-

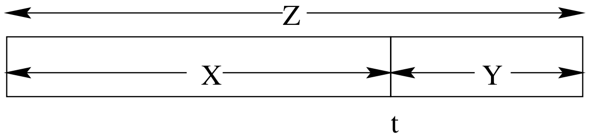

Figure 4.1. Two neighboring windows with acoustic vector sequences X and Y around time t, when we want to decide if a change point exists or not.

mates of the parameters of these Gaussian PDFs (the means and standard deviations) are denoted as θ_x, θ_y and θ_z, respectively in this chapter.

Various metric based algorithms differ in the kind of dissimilarity function they employ, the size of the two windows, the time increments of the shifting of the two windows, and the way the resulting dissimilarity values are evaluated and thresholded. However, the most important research issue in this framework is the choice of the dissimilarity function and the way it is used to make the decision about a potential change point. The next section explains some of the most commonly used criteria for this purpose.

4.3 Common criteria

4.3.1 Log Likelihood Ratio (LLR)

The problem of detecting a speaker change point as defined in previous section can be formulated as a hypothesis test where the null hypothesis H_0 is that there is no speaker change at time t or the segments X and Y were generated by the same speaker. The log likelihood of this hypothesis given segments X and Y, L_0, is given as:

$$L_0 = \sum_{n=1}^{N_x} \log p(x_n|\theta_z) + \sum_{n=1}^{N_y} \log p(y_n|\theta_z) \tag{4.1}$$

where, as mentioned earlier, θ_z denotes the ML parameters of the Gaussian density estimated over complete dataset Z.

On the other hand, the alternate hypothesis H_1 is that there exists a change point at time t or the segments X and Y were generated by different speakers. The log likelihood of this hypothesis, L_1, given segments X and Y is given as:

$$L_1 = \sum_{n=1}^{N_x} \log p(x_n|\theta_x) + \sum_{n=1}^{N_y} \log p(y_n|\theta_y) \tag{4.2}$$

where θ_x and θ_y are the parameters of individual Gaussian densities of segments X and Y, respectively.

Note that although L_1 (4.2) is mentioned in previous literature (*e.g.* (Gish *et al.*, 1991)) as log likelihood of the data under the hypothesis H_1, it is not true. In absence of a valid PDF for hypothesis H_1, L_1 is not a valid log likelihood.

Following the hypothesis test decision rule in (2.7), a decision to "reject H_0 in favor of H_1" or to decide a speaker change point at time t is made, if:

$$d_{llr} = L_0 - L_1 \leq Threshold \tag{4.3}$$

where d_{llr} is the resulting LLR metric between the two windows.

Since, the 'maximum' likelihood parameters for subsets X and Y in hypothesis H_1 are estimated individually, each of the two terms on right hand side of (4.2) are greater than the corresponding term in (4.1). Hence we always have $L_1 \geq L_0$ or $d_{llr} \leq 0$. Thus, a threshold is needed to make a decision about a speaker change point using this metric. This threshold is generally set empirically, mostly using a development dataset and needs to be adjusted for different test conditions.

4.3.2 Bayesian Information Criterion (BIC)

It was mentioned in Chapter 2 that BIC is a model selection criterion and was used for speaker change detection and clustering by Chen and Gopalakrishnan (1998) who formulated this problem as a model selection problem. Chen and Gopalakrishnan (1998) compared two models: one models the data as just one Gaussian (no speaker change) and the other (a speaker change) models the data as two Gaussians, one for each of the dataset X and Y. The difference of the BIC values (2.8) of the two models can be written as:

$$d_{bic} = L_1 - L_0 - \frac{\lambda}{2} \cdot \Delta K \cdot \log N \tag{4.4}$$

where L_1 (4.2) and L_0 (4.1) are the maximum log-likelihoods of the two models as defined in previous section, N is the total number of data-points in Z, ΔK is the difference in the number of free parameters in the two models and λ is the penalty weight. The comparison of (4.4) and (4.3) shows that BIC based metric d_{bic} is essentially a penalized (thresholded) LLR, where the penalty (threshold) depends on the difference in the number of free parameters in the two models.

Note that the original BIC criterion (2.8) proposed in (Schwarz, 1978) does not have the parameter λ. It was introduced by Chen and Gopalakrishnan (1998) as a "tuning" parameter to obtain various degrees of segmentation and clustering performance.

Another point to note here is that in this model selection formulation of the problem, the model where the data is modeled as two Gaussians is not valid. This model does not have a PDF and hence the estimation of the maximum log likelihood of the model, L_1 (4.2) is not a valid log likelihood, as required by true BIC (2.8). This is similar to the problem mentioned in the previous section in the context of hypothesis testing.

BIC is supposed to have the advantage of not requiring any thresholding. Ideally, local maxima of d_{bic} greater than 0 should be speaker change points. However, this is only true if $\lambda = 1$ or if there is a systematic way to find the optimal value of λ. In absence of this, λ is an implicit threshold embedded into the penalty term. This fact has been mentioned in previous work and was also noticed during our experiments as discussed in Section 4.5. In (Kemp *et al.*, 2000), it was mentioned that the threshold found using the BIC principle (with $\lambda = 1$) yielded significantly worse results compared to the best possible threshold selection. In (Tritschler and Gopinath, 1999; Vandecatseye and Martens, 2003), the value of λ used was different than 1.0. In (Delacourt and Wellekens, 2000) a development dataset was used to find the optimal value of this parameter.

It is mentioned in (Chen and Gopalakrishnan, 1998) and we observed during our experiments that a higher value of λ fails to detect many genuine speaker changes and a lower value, on the other hand, results in many false alarms.

4.4 Proposed Criterion

In our approach, we follow the same notations and similar framework defined in previous section. However, we do not refer to it as hypothesis testing or model selection problem for the reason that was mentioned in Sections 4.3.1 and 4.3.2 that L_1 is not a valid log likelihood. In the proposed criterion, we calculate L_1 in the same way (4.2), but rather refer to it as a score.

For the proposed criterion, we define another score, L_0', which is indeed the log likelihood of a GMM with 2 Gaussian components, given the complete dataset Z. The ML estimates of the parameters of this GMM, θ_z', are calculated using EM algorithm. L_0' is then calculated as:

$$L_0' = \sum_{n=1}^{N_x} \log p(x_n|\theta_z') + \sum_{n=1}^{N_y} \log p(y_n|\theta_z') \tag{4.5}$$

Since a GMM can always fit the data better (or equally well) compared to a single Gaussian density, we always have:

$$L_0' \geq L_0 \tag{4.6}$$

where L_0 (4.1) was defined in the previous section. The proposed distance metric d_{llrc} is then simply the difference of the two scores L_1 (4.2) and L_0' (4.5):

$$d_{llrc} = L_1 - L_0' \tag{4.7}$$

Figure 4.2 shows the behavior of the proposed distance metric (when used in a framework explained in Section 4.5) on a sample waveform consisting of 3 speaker turns. Figure 4.2 shows that the local maxima of the proposed distance metric greater than zero (the zero is shown by the horizontal line) clearly identify these true change points (indicated by stars in the figure).

From (4.7), d_{llrc} is very similar to LLR metric d_{llr} as given by (4.3). However, unlike d_{llr} which is always negative, the proposed metric ranges from positive to negative values, making zero a natural threshold for this problem. All local maxima of d_{llrc} greater than zero are considered to be speaker change points. It can be seen that, in contrast to the standard LLR and BIC techniques,

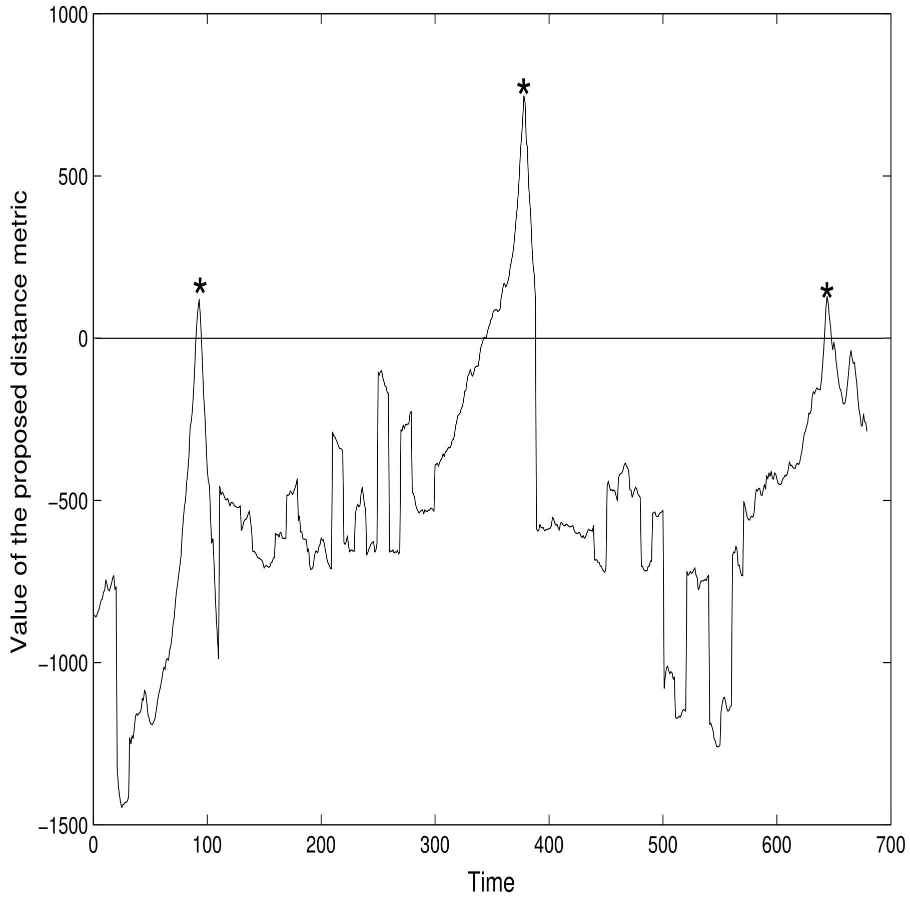

Figure 4.2. The behavior of the proposed distance metric on a sample speech consisting of 3 speaker turns, as implemented using the algorithm mentioned in Section 4.5. The horizontal line in the figure shows the zero level of the distance metric, and the stars indicate the location of true change points. The figure clearly shows that the local maxima of the proposed distance metric greater than zero correspond to the true change points.

all terms in this criterion (4.7) are derived directly from the data, and thus the criterion can be expected to be robust to changing data conditions. To get a further insight into the working of this criterion, let us consider two extreme theoretical cases.

Case 1: Let us assume that the subsets X and Y come from two very different speakers and hence have very distinct PDFs such that:

$$p(x_i|\theta_y) \ll p(x_i|\theta_x) \qquad \forall x_i \in X \tag{4.8a}$$

$$p(y_i|\theta_x) \ll p(y_i|\theta_y) \qquad\qquad \forall y_i \in Y \tag{4.8b}$$

Or, in other words we assume that a model with parameter θ_x is a better match to every point in dataset X compared to the model with parameters θ_y and vice-versa. In this case the parameters of the two Gaussian components of the GMM (estimated by EM algorithm) would approximately converge to θ_x and θ_y, and their weights (w_1 and w_2) will approximately converge to $\frac{N_x}{N}$ and $\frac{N_y}{N}$, respectively. Based on this, L_0' can be written as:

$$L_0' \approx N_x \log w_1 + \sum_{i=1}^{N_x} \log p(x_i|\theta_x) + N_y \log w_2 + \sum_{i=1}^{N_y} \log p(y_i|\theta_y) \tag{4.9}$$

Using (4.9) and (4.2) in (4.7), d_{llrc} can be written as:

$$d_{llrc} \approx -N_x \log \frac{N_x}{N} - N_y \log \frac{N_y}{N} \tag{4.10}$$

From (4.10), it follows that in this extreme case:

$$0.0 < d_{llrc} \leq N \log 2.0 \tag{4.11}$$

Since $d_{llrc} > 0$, the proposed criterion will favor a change point at time t.

Case2: Let us assume that X and Y come from the same speaker, and have very similar densities. In an extreme case, we can assume that $\theta_x \approx \theta_y \approx \theta_z$ (θ_z parameterize the Gaussian density of complete data Z). This implies that $L_1 \approx L_0$. This together with (4.6) also implies that $d_{llrc} \leq 0$ and hence the proposed criterion will not detect a speaker change at time t. The equality sign in this case is only of theoretical interest and holds in a limiting case when the *real* PDFs of X and Y are mono-Gaussians and exactly the same, which does not occur in practice.

Based on this, it can be seen that more negative values of this criterion provide a strong confidence toward no speaker change and, on the other hand, more positive values provide high confidence toward potential speaker change points. Of course, it may still be advantageous to tune (whenever possible) an explicit threshold term to further improve performance on a given dataset.

However, the principal advantage of this approach is that it should provide robust performance even without tuning any parameter.

4.5 Experimental Setup

An experimental setup was designed (Ajmera *et al.*, 2004b) to compare the performance of the proposed criterion with the main state-of-the art approach for this purpose, the BIC. It was mentioned earlier that the speaker change detection framework also involves shifting and resizing of the two neighboring windows. We basically adopted the algorithm presented in (Chen and Gopalakrishnan, 1998) for this purpose while also implementing some useful tips presented in (Tritschler and Gopinath, 1999). The algorithm is as follows:

```
1. initialize the interval [a, b]
     a =0, b = MIN_WINDOW;
2. find the change point in [a, b] according to a criterion.
     for BIC find the point of local maxima of $d_{bic} > 0$;
     for proposed criterion, find the point of local maxima of $d_{llrc} > 0$.
3. if no change in [a, b]
      b = b+ MORE_FRAMES;
   else if 't' is the change point in [a, b]
      a = t+1, b = a + MORE_FRAMES;
4. if b-a > MAX_WINDOW
      a = b - MAX_WINDOW;
5. go to (2)
```

where, the values like `MORE_FRAMES`, `MAX_WINDOW` etc. are also optimized for good performance as well as keeping the computational complexity reasonable to allow real-time implementation of the algorithm. In our experiments `MORE_FRAMES` and `MAX_WINDOW` were chosen to correspond to 1 second and 10 seconds of speech, respectively. For comparison of the proposed approach with BIC, the basic framework was kept unchanged apart from the criterion in step 2 above. 12-MFCC coefficients (without C_0 and derivatives) were used as acoustic feature vectors extracted every 10ms.

The choice of these feature vectors was based on the comparison across a set of feature vectors made in Section 5.5.1. Diagonal covariance matrices were used for all the experiments

The HUB-4 1997 evaluation set was used to test the performance of the proposed criterion. The HUB-4 database consists of nearly 3 hours of broadcast news data in different acoustic conditions, totaling 515 speaker changes from a large variety of speakers (total 117 speakers and other non-speech segments), and thus gives a good test of robustness.

4.5.1 Evaluation Metric

A change detection system has two possible types of error. Type-I errors occur if a true change is not spotted within a certain window (1 second in our case). Type-II errors occur when a detected change does not correspond to a true change in the reference (false alarm). Type I and II errors also correspond to as recall (RCL) and precision (PRC) respectively, which are defined as:

$$PRC = \frac{\text{number of correctly found changes}}{\text{total number of changes found}} \tag{4.12a}$$

$$RCL = \frac{\text{number of correctly found changes}}{\text{total number of correct changes}} \tag{4.12b}$$

In order to compare the performance of different systems, the F-measure is often used and is defined as:

$$F = \frac{2.0 * PRC * RCL}{PRC + RCL} \tag{4.13}$$

The F-measure varies from 0 to 1, with a higher F-measure indicating better performance.

4.5.2 Results

The results using the proposed criterion and the BIC are presented in Table 4.1. This table shows that the parameter λ acts as a threshold in (4.4). A lower value of λ (less than 5 in this case) results in detecting most of the reference change points correctly *i.e.* high RCL, but also results in high number of false alarms *i.e.* low PRC. A higher value on the other hand (greater than 7 in this case) results in low RCL and high PRC. The table also shows that the performance of the proposed merging criterion (4.7) when used in speaker change detection framework, is comparable (in fact better, even if slightly) to that of BIC with optimal value of $\lambda = 6.0$ chosen.

However, the important property of the proposed criterion is that all the terms involved (4.7) in making a decision about a change point are based on the behavior of the current data and the models based on it. Thus, the criterion is expected to adapt to different data conditions automatically. Conversely, the threshold value in LLR and the penalty term in BIC is based on the behavior of a different dataset (development data) and may not apply across datasets having different environmental conditions, speaker characteristics etc.

To further verify this, we used a 30-minute broadcast news development dataset to find the optimal value of λ. The optimal value of λ for this dataset turned out to be 5 (resulting in $F = 0.57$), which is different from the optimal value of $\lambda = 6$ for the test data (Hub4 1997 evaluation set). Moreover, $\lambda = 6$ resulted in a sub-optimal performance ($F = 0.55$) for the development data. This shows that the optimal value of λ is different from dataset to dataset and that the performance of BIC as used for speaker change detection is highly dependent on the choice of this parameter.

Criterion	RCL	PRC	F
Proposed	0.65	0.63	0.64
BIC ($\lambda = 1.0$)	0.79	0.22	0.34
BIC ($\lambda = 4.0$)	0.78	0.45	0.57
BIC ($\lambda = 5.0$)	0.72	0.55	0.62
BIC ($\lambda = 6.0$)	0.67	0.60	0.63
BIC ($\lambda = 7.0$)	0.60	0.63	0.62
BIC ($\lambda = 8.0$)	0.51	0.65	0.57

Table 4.1. Results of the proposed criterion and BIC (with different values of λ) on HUB-4 1997 evaluation data.

Apart from comparing the performance of the proposed criterion to that of BIC as shown in Table 4.1, we also analyzed the errors made by the system in detail. There are 515 true speaker changes in reference. Our system detected 550 changes, out of which 332 changes corresponded to the true change points. This resulted in total 220 insertions and 183 deletions (with 1 second tolerance window). In our analysis, we found that there are 44 segments when multiple speakers are speaking (overlapping) over a window of less than 2 seconds. During these segments, it is difficult even for a human listener to define boundary points. The data also consists of a large number (66, excluding multiple speaker segments) of speaker change points with *detectability* less

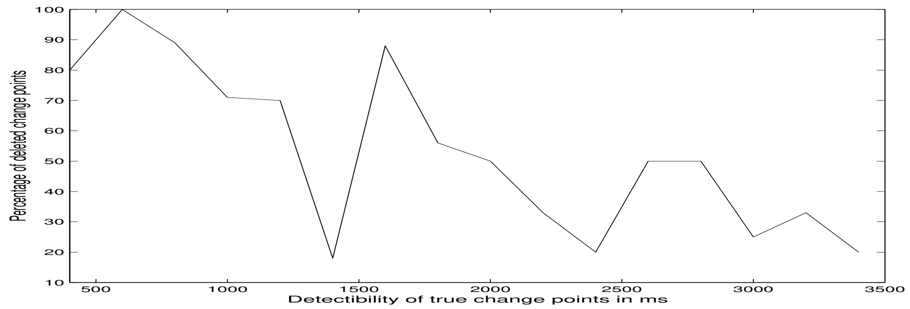

Figure 4.3. The histogram of deleted true reference speaker change points as a function of their detectability. The detectability of a speaker change point is defined as the duration of the segment following that change. The figure shows that the accuracy of the proposed algorithm depends on the detectability of true change points.

than 2 seconds, where the detectability of a reference change point is defined as the duration of the segment following that change. Figure 4.3 shows the histogram of deleted speaker turns with respect to the detectability of reference speaker changes. This figure shows that the performance of the proposed speaker change detection framework for a reference change point depends a lot on its detectability.

26 insertions were made during pure music regions. Moreover, 16 insertions correspond to the true change points slightly deviated (less than 2 seconds) from their true position. Figure 4.4 shows the biases of the detected change points from the true reference change points. From Figure 4.4, it can be seen that the proposed speaker change detection framework detects most of the reference changes within a very small range (nearly 80% of the change points are detected within 400ms of the corresponding true reference points). Ignoring the multiple speaker segments, insertions during music regions and slightly deviated change points from scoring, we obtain the results presented in Table 4.2.

Criterion	RCL	PRC	F
Proposed	0.73	0.66	0.69
BIC ($\lambda = 6.0$)	0.73	0.61	0.67

Table 4.2. Results of the proposed criterion and BIC (with $\lambda = 6$) after ignoring multiple speaker segments, insertions made during music regions and slightly (less than 2 seconds) deviated change points.

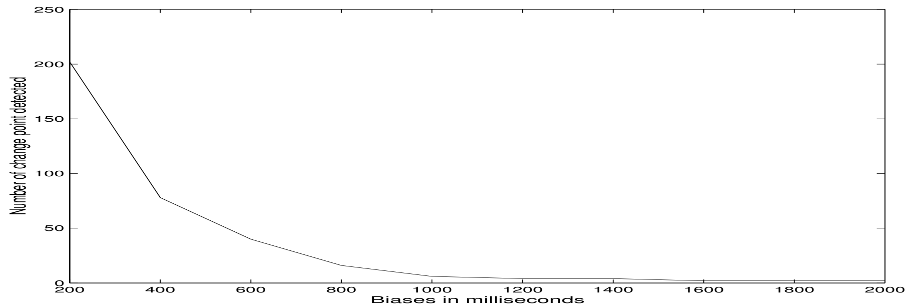

Figure 4.4. Histogram of the biases of the detected change points with respect to the location of the true reference change points. The figure shows that most of the detected change points are precisely located with respect to the corresponding true reference speaker boundaries. by the proposed speaker change detection framework. Nearly 80% of the detected change points are within 400ms of the corresponding true reference points

4.6 Conclusion

This chapter addressed the issue of detecting speaker turns in a continuous audio stream. To achieve this, first a general framework which is used by most of the metric based approaches is explained. A detailed study of the most common distance metrics (LLR and BIC) for this purpose was presented, mainly, to understand why a threshold value is required to make decisions about speaker turns while using these criteria.

Based on the understanding of these criteria and their limitations, a novel distance metric for speaker change detection was proposed in this chapter. In contrast to other metric based approaches, the proposed criterion does not require an adjustable threshold/penalty value to make decisions about speaker turns. The form of proposed distance metric is similar to that of LLR except that the values of the proposed metric range from negative to positive, making zero a natural threshold. An intuitive and systematic explanation of the functioning of the proposed criterion was also provided. The usefulness and robustness of this criterion was further illustrated with the help of experiments where the proposed criterion achieved comparable results to that of using BIC with optimal tuning of penalty term.

The detailed analysis of the results indicates that the accuracy of the proposed framework depends on the detectability (duration of the segment following that change) of true speaker change points. Generally, segments of detectability less than 2 seconds are not detected accurately by the system. Also, the system finds multiple change points during pure music regions. However, most of the significant (detectability greater than 2 seconds, clean speech) true change points are detected precisely (within a range of 400ms) by the proposed speaker change detection framework.

When used in an online framework, the proposed criterion makes a robust and unsupervised tool to segment the audio stream in terms of homogeneous regions. This framework is further used in Chapter 6 as part of an audio indexing application. It should be noted that such a segmentation is also appropriate as a preprocessing module for an ASR system to convert the continuous audio stream in terms of manageable and homogeneous chunks. The proposed SCD framework is presently being used for Mandarin broadcast news ASR at ICSI[1] (Stolcke *et al.*, 2003).

[1] International Computer Science Institute, http://www.icsi.berkeley.edu

Chapter 5

Speaker Clustering and Segmentation

5.1 Introduction

The “clustering” process in general can be defined as unsupervised classification of data *i.e.* without any *a priori* knowledge about the classes or the number of classes. There are many instances where classification must and can be performed without *a priori* knowledge. Consider, for example, the biological taxonomy problem. Over the years, all known living things have been classified according to certain observable characteristics. Of course, plants and animals have never borne labels indicating their kingdom, phylae and so on. Rather, they have been categorized according to their observable characteristics without outside supervision.

The clustering problem is not well defined unless the resulting classes are required to exhibit certain properties. The choice of properties or, equivalently, the definition of a cluster, is the fundamental issue in the clustering problem. Given a suitable definition of a cluster, it is possible to distinguish between a good and bad classification of samples.

In the case of speaker clustering, the definition of the task is not very difficult. Ideally, the clustering should result in a single cluster for every speaker. However, in applications like speaker adaptation, it may be desirable to have several big clusters of speakers, where each cluster has data belonging to speakers whose acoustic properties (this may come from gender, speaking style, speaking rate, etc.) are very similar. Thus, depending on the application, there can be different ways of evaluating a clustering solution. For example, in the speaker adaptation paradigm, the improvement in the recognition accuracy accrued by the adaptation process is a good evaluation criterion. However, to get an objective evaluation across all the applications, we can define a good clustering solution when there is perfect one-to-one mapping between the speakers and their clusters in the solution.

Speaker clustering is important for many practical applications. It has been repeatedly shown that speaker adaptation results in significant improvement in ASR accuracy (Chen and Gopalakrishnan, 1998; Johnson and Woodland, 1998), and speaker clustering is an essential step toward speaker adaptation. A by-product of the speaker clustering process is a segmentation in terms of speakers, which can be useful in many audio indexing applications and meeting summarization.

As already mentioned in Chapter 2, there has been a lot of previous work on this problem. Some of the previous work assumed a known number of speakers, which is not generally the case in many practical situations. In some cases, segmentation in terms of either utterances or speakers was assumed, leaving the solution not very practical. Also, some work reported an independent segmentation step prior to the clustering stage and our criticism of this approach is that the errors made in the segmentation step can not be recovered in the later stages.

A major component of the problem of any clustering is finding optimal number of clusters (N_c). In practice, the number of classes in which the data should be clustered is rarely known, and has to be found as part of the problem. For this purpose, one may run the clustering procedure for different number of clusters, and find the best classification for each of these values. If the optimality criterion decreases as number of clusters increases, and either reaches the minimum point at some value of N_c or becomes flat after some value of N_c, we may use N_c as the proper number of classes. Unfortunately, many of the popular criteria do not have this favorable property. It appears, therefore, that some external method of controlling N_c is necessary. The biggest limitation of most of the previous approaches for controlling N_c, and as clearly pointed out in Chapter 2 is that either a heuristic (such as a pre-meditated number of clusters or minimum occupancy criterion) is applied or a threshold-like term is used to ensure a good stopping point. These heuristics or threshold values are decided empirically and vary from situation to situation. Thus, the clustering solution is no longer robust to unseen data conditions or a development dataset is required to adapt the system to the unseen data.

In this chapter, a speaker clustering algorithm is presented which takes into consideration the above mentioned limitations of the previous work. The clustering solution is achieved in an ergodic HMM framework (similar to the framework in Chapter 3 but with a greater number of classes), where each state in the HMM represents a cluster. In the absence of any *a priori* information about the number of speakers, the algorithm starts with over-segmentation, *i.e.* a large number (significantly greater than expected number of speakers) of clusters. The standard EM algorithm (Dempster *et al.*, 1977) is employed to train the HMM in an iterative manner. After this initial HMM topology is trained, iterative merging of appropriate clusters is done until there are no more candidates left for merging. A novel distance metric and a novel merging and stopping criteria based on this metric are also proposed for this purpose. The important point about this metric is that it is not a monotonic function of the number of clusters and it thus defines a unique stopping point for the criterion. Furthermore, all this is achieved without applying any heuristics or adjustable thresholds.

The chapter is organized as follows: Section 5.2 explains the speaker clustering algorithm and different modules associated with this. This section also clearly points out several issues associated with an agglomerative clustering framework, especially the need for a merging and stopping criteria. Section 5.3 first explains the use of LLR as a merging criterion and then points out the need for thresholding as a stopping criterion while using LLR in Section 5.3.1. Section 5.3.2 presents the proposed threshold free stopping criterion, while using LLR as a merging criterion. Building on these two merging and stopping criteria, Section 5.3.3 explains the proposed merging criterion, which in turn also leads to a threshold free method of controlling the clustering algorithm. Section 5.4 presents the evaluation metric and data used to validate parameters of the proposed algorithm and also to compare its performance against other approaches. Section 5.5 presents a detailed analysis of roles of different system settings and their effect on the system performance. In Section 5.6 we analyze and discuss the performance of the proposed algorithm and also compare it with the performance of LLR as a merging criterion. Section 5.7 discusses our participation in NIST[1] evaluations, where the performance of the proposed system was compared with other state-of-the-art

[1] National Institute of Standards and Technology, http://www.nist.gov/

approaches in this domain.

5.2 Speaker Clustering Algorithm

Figure 5.1 presents an overview of the speaker clustering algorithm. The clustering algorithm is based on an ergodic HMM formalism where the number of states in the HMM is equal to the initial number of clusters. Figure 5.2 presents the topology of this HMM. Each state in this HMM is composed of several sub-states. These sub-states impose a minimum duration constraint on the states of the model. Each state of the HMM represents a cluster and is finally expected to represent a single speaker. The PDF of each state is assumed to be a GMM with M Gaussian components, which are shared among all sub-states. Thus, in the following, a cluster will mean a set of data points and the associated PDF, modeled by a GMM with M Gaussian components.

The individual modules of the algorithm shown in Figure 5.1 are explained in the following sections.

5.2.1 Initialization

In the absence of any *a priori* information about the number of speakers (and hence number of clusters), we start by over-segmentation of the data. "Over-segmentation" refers to the process of segmenting the data into an excessive (compared to the expected number of speakers) initial number of clusters K. The HMM topology thus, has K states fully connected to each other. The parameters of this HMM are the initial probabilities, the transition probabilities and the parameters of the PDFs (GMM in this case) of all the states of the HMM. In our experiments, we noted that the transition probabilities do not play a big role in the clustering process and its performance, and are thus set manually. In the absence of any *a priori* information, the initial probabilities of all the states were also set uniformly. The initialization of the GMM parameters is either done assuming uniform segmentation in the beginning or using the K-Means algorithm explained in (Rabiner and Juang, 1993). Further details about these two initialization schemes and their comparative performance are given in Section 5.5.2.

After this initialization, the parameters of the HMM are trained in an unsupervised manner via the EM algorithm. This is an iterative process, consisting of a segmentation step (explained in

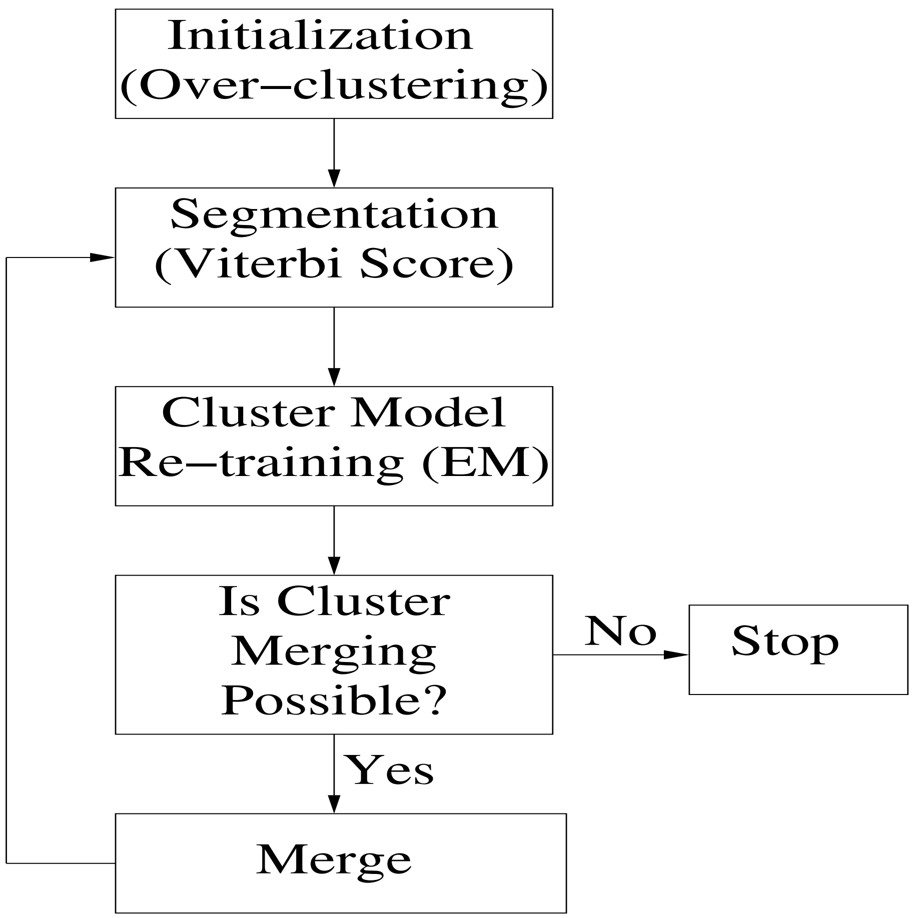

Figure 5.1. The overall description of the speaker clustering algorithm.The HMM based algorithm runs in an agglomerative fashion, starting with large number of clusters. Most similar clusters (decided by proposed similarity criterion) are merged in an iterative manner, until there is no more suitable pair of cluster candidates. The segmentation in terms of these clusters is also obtained before every update of system parameters.

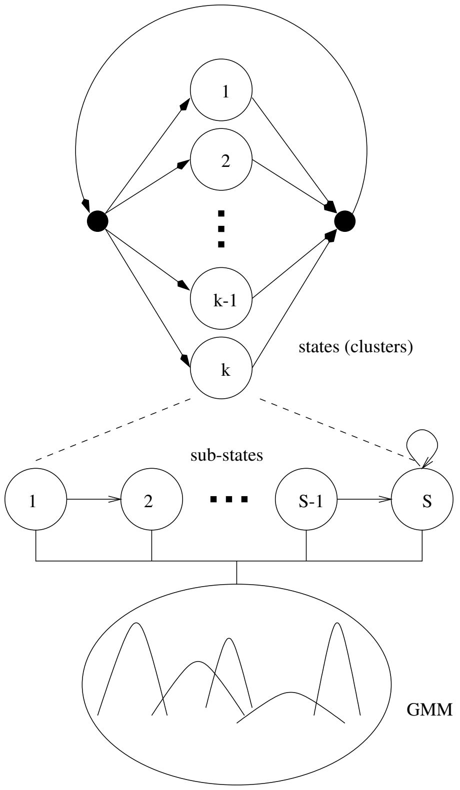

Figure 5.2. HMM Topology used for speaker clustering. Every speaker hyper-state is composed of several sub-states, ensuring minimum duration. All sub-states of a speaker class share a common PDF modeled by a GMM.

Section 5.2.2) using current set of parameters and then updating (re-training) the parameters of all the clusters (explained in Section 5.2.3) based on this segmentation to maximize the likelihood of the data. The initial HMM topology with K clusters (states) is thus trained iteratively, resulting in over-segmentation of the data.

A result of over-segmentation is that data from a single speaker is likely to be assigned to different clusters. At the same time, it is desirable that data from two speakers are not assigned to one cluster. Following an agglomerative clustering framework, the goal is then to find clusters which are best candidates for merging (clusters having data from same speaker) and this is achieved in subsequent stages of the algorithm.

5.2.2 Segmentation

The well known Viterbi algorithm is used to segment the data based on the current model parameters (Λ). The Viterbi segmentation tries to find the best possible state sequence, q_{best}, under given constraints (minimum duration in this case) such that the joint likelihood of the data and this sequence (Viterbi score) given current set of parameters, Λ, is maximized *i.e.*,

$$q_{best} = \arg\max_{q} p(X, q|\Lambda) = \arg\max_{q} p(X|q, \Lambda)p(q|\Lambda) \tag{5.1}$$

5.2.3 Cluster Models Re-Training

Based on the segmentation achieved in the previous step, a set of data is assigned to each cluster. The parameters of the PDF (GMM in this case) of each cluster are then individually re-trained using the EM algorithm for Gaussian mixture densities. The EM algorithm updates the parameters of the GMM in an iterative manner to maximize the likelihood of the data in each cluster given the parameters of the model (GMM) of that cluster. This results in an updated set of parameters (Λ') for the HMM maximizing the joint likelihood of the overall data sequence X and best possible path sequence, q_{best} (Viterbi score), as follows:

$$\Lambda' = \arg\max_{\Lambda} p(X, q_{best}|\Lambda) \tag{5.2}$$

where, q_{best} is the solution of Eq. 5.1.

After iterative segmentation and cluster model re-training steps, the initial HMM topology is trained and the data X is over-clustered. A consequence of over-segmentation is that data belonging to some speakers is split across a number of clusters. Thus, the next step in the speaker clustering process is to identify such clusters and merge them as explained in the following section.

We note that the re-training of cluster models is also possible using full EM (generally known as Baum-Welch algorithm (Rabiner and Juang, 1993) for the HMMs), without making a big difference, especially because of the minimum duration. However, full EM training is more computationally expensive than the Viterbi approximation used in this thesis.

5.2.4 Cluster Merging

Cluster merging refers to the operation of merging the data associated with two clusters and assigning it to a single cluster, and the parameters of this resulting cluster are trained to maximize the likelihood of the combined dataset. In an agglomerative clustering framework, this is generally done in an iterative fashion. There are three important issues to be considered regarding this:

- **How to select appropriate candidates for merging?**: This typically involves employing a similarity metric between all possible pair of clusters and choosing the one with minimum/maximum value of this metric. One example of such similarity metric is the log-likelihood ratio (LLR) which is discussed later in this chapter.

- **How to merge the two clusters?**: After two clusters are merged and the data is assigned to a single cluster, the HMM topology changes. In the HMM framework discussed so far, it is clear that after one such merging, the HMM topology would change to have one less state and accordingly the number of trainable parameters in HMM would change.

- **When to stop merging**: Most of the popular objective criteria used in the literature either monotonically increase or decrease with merging of two clusters in each iteration. Consider, for example, the Viterbi score on the right hand side of (5.2). With every merging of two clusters in an iteration, if the total number of parameters decreases (*i.e.* $|\Lambda_k| < |\Lambda_{k-1}|$), the Viterbi score will also decrease monotonically *i.e.*,

  $$p(X, q_{best,k}|\Lambda_k) < p(X, q_{best,k-1}|\Lambda_{k-1}) \tag{5.3}$$

  where the subscript k denotes the current number of clusters, Λ_k denotes the parameters and $q_{best,k}$ denotes the best segmentation of the data in terms of these k clusters.

  Thus, it is difficult to decide when to stop the iterative merging process and an external method of stopping this process is necessary. It is possible to apply heuristics in *e.g.* the merging process is stopped after having obtained pre-determined number of clusters or, occupancy of a cluster goes beyond a limit (Johnson and Woodland, 1998). Another solution is to compare the decrease or increase in the objective function against a threshold (Siegler *et al.*, 1997). Generally, these threshold values are found empirically and point where the value of objective function goes beyond this threshold is considered to be the stopping point.

The next section discusses these issues in detail and also presents the proposed similarity metric, which provides a threshold and heuristic free solution to all the three above mentioned problems.

5.3 Merging and Stopping Criteria

5.3.1 Thresholded Log Likelihood Ratio

For the purpose of calculating similarity metric between two clusters C_a and C_b with data D_a, D_b and PDFs parameterized by θ_a,θ_b respectively, let us formulate the problem as a hypothesis test. The null hypothesis H_0 is that data D_a is from cluster C_a and the alternate hypothesis H_1 is that the data D_a is from cluster C_b. Computing the ratio of log likelihoods of these two hypotheses and making it symmetrical[2] with respect to the two clusters will result in:

$$d_{llr}(C_a, C_b) = \log \frac{p(D_a|\theta_a)}{p(D_a|\theta_b)} + \log \frac{p(D_b|\theta_b)}{p(D_b|\theta_a)} \tag{5.4}$$

The first term on the right hand side of Eq. 5.4 is an indication of how well the PDF of cluster C_b can model the data associated with cluster C_a. The best case for merging is when the PDF of cluster C_b can model the data associated with cluster C_a equally well and in that case this term would become zero. From (5.4), the value of d_{llr} is always nonnegative, and that the smaller the value of d_{llr}, the better the candidates for merging. This value has 0 as lower limit and theoretically no upper limit. Thus, if LLR is to be used as a similarity metric for merging criterion, it has to be compared against a threshold value to decide if the two clusters should be merged or not.

The stopping criterion in this case is when the lowest similarity measure among pairs of clusters given by (5.4) goes beyond this threshold value. This merging and stopping criterion was used in our experiments and the results and issues associated with this are reported in Section 5.6.2

[2]We note that after this symmetrization, the resulting metric is not LLR but we associate the term LLR with the distance for convenient reading

5.3.2 Log Likelihood Ratio and Proposed Stopping Criterion

The fact that the use of LLR requires finding a threshold value to ensure a good stopping point is a result of behavior explained by (5.3). This happens if the total number of parameters used to model the data is also reduced after every pair of clusters and hence this decrease in the Viterbi score is not surprising. However, if we force the total number of parameters to be constant even after merging, we observe that the Viterbi score no longer decreases monotonically. This was reported in (Ajmera *et al.*, 2002) and is explained below:

- Two clusters with minimum LLR according to (5.4) are selected for merging. Let these clusters be C_a and C_b.
- A cluster C is hypothesized as a result of merging the two clusters. The PDF (a GMM) of this cluster is modeled by using same number of parameters as the sum of the number of parameters in the individual PDFs of C_a and C_b. For example, if the two individual clusters (C_a and C_b) have M_a and M_b Gaussian components in their GMMs, the hypothesized cluster is modeled by a GMM with $M_a + M_b$ number of Gaussian components. One of the states associated with the two clusters C_a and C_b is now assigned this PDF and the other is omitted from HMM topology. We note that although the HMM topology has now changed, the total number of trainable parameters remains constant.
- Under these constraints, the Viterbi score first increases and then starts decreasing. We observed that if we consider this point where the Viterbi score starts decreasing as the stopping point, the resulting clustering performance was quite comparable to the system performance when a threshold is used for stopping criterion and this is shown in Section 5.6.2. This also has the obvious advantage that the stopping criterion is met without using any adjustable threshold value.

In our analysis, it was observed that in the next segmentation (from $q_{best,k}$ to $q_{best,k-1}$), the only segmentation changes were due to cluster merges and the remaining boundaries moved very little. Moreover, the analysis showed that the Viterbi score was increasing for every good merge *i.e.* merging of two clusters having data from the same speaker. The Viterbi score started decreasing after these “good” candidates were over and remaining candidate pairs were merged. Based on these observations, a novel merging criterion was proposed in (Ajmera and Wooters, 2003) and is explained in Section 5.3.3.

5.3.3 Proposed Merging and Stopping Criterion

We define our merging criterion as follows:

- Consider a pair of clusters C_a and C_b with datasets D_a and D_b respectively.
- Let M_a and M_b represent the number of Gaussian components in the PDF of two clusters parameterized by θ_a and θ_b, respectively.
- Let us hypothesize a new cluster C having data $D = D_a \cup D_b$ with a PDF modeled by a GMM parameterized by θ with $M_a + M_b$ number of Gaussian components. Also, the parameters of this hypothesized cluster are trained via EM algorithm to maximize the likelihood of complete dataset D.

Given these conditions, the pair of clusters (C_a and C_b) becomes a candidate for merging if the following is true:

$$\log p(D|\theta) \geq \log p(D_a|\theta_a) + \log p(D_b|\theta_b) \tag{5.5}$$

Criterion in (5.5) can also be expressed as:

$$\log \frac{p(D_a|\theta_a)}{p(D_a|\theta)} + \log \frac{p(D_b|\theta_b)}{p(D_b|\theta)} \leq 0.0 \tag{5.6}$$

From (5.6), it is evident that the proposed merging criterion also has the form of LLR, with a difference that the denominator now is a likelihood associated with the hypothesized cluster. This makes the most important difference to LLR (which is always positive), since the value of the term on the left hand side of Eq. 5.6 is not always positive or negative. We select the pair for merging only if this quantity is negative. Another way of looking at this is by saying that this is an LLR similarity measure with zero threshold. This is also similar to BIC methodology (2.8) of model selection where

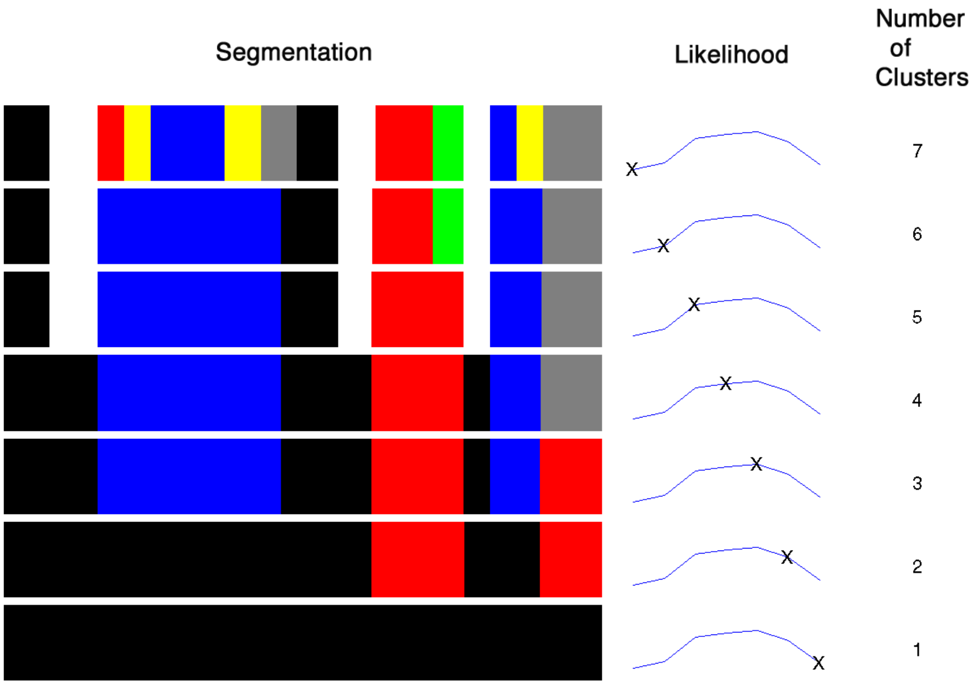

Figure 5.3. The figure shows the segmentation, resulting Viterbi score and associated number of clusters after every iteration in proposed speaker clustering algorithm. In this example, there are 3 speakers, and the algorithm starts with 7 clusters. After every iteration of merging two clusters, the Viterbi score increases until there are 3 clusters left and then starts decreasing. The algorithm is stopped at the point when this Viterbi score starts decreasing. This shows that the algorithm tries to find maxima of Viterbi score given a fixed number of parameters.

the number of free parameters in the two models are equal, making the log likelihoods of the two models directly comparable.

It was observed consistently that selecting candidates for merging using this criterion, while keeping the number of parameters in the two models equal, always results in an increase in the Viterbi score in the next iteration. In other words, in this framework, we are trying to maximize the Viterbi score of the data given a fixed number of parameters. This phenomenon is shown in Figure 5.3. Moreover, it is shown in Section 5.4 that using the proposed merging criterion always results in a better performance compared to when LLR is used as a merging criterion in both thresholded (Section 5.3.1) and proposed framework (Section 5.3.2).

After every new segmentation-training step, we look for the best pair of clusters satisfying (5.5). In the case of many such candidate pairs, we choose the pair that maximizes the difference of the terms of the left hand side and right hand side of (5.5). The merging is stopped when there are no suitable candidates satisfying (5.5).

5.4 Evaluation Metric and Data

In this section, we explain the evaluation metric and data that is used to evaluate the performance of the algorithm with different settings and also to compare the performance of proposed algorithm with other approaches.

5.4.1 Evaluation Metric

The concept of cluster purity, introduced in (Solomonoff *et al.*, 1998), is an indication of how well the data assigned to a cluster is limited to a particular speaker in the reference. However, it was realized that this is not enough to measure the performance of the clustering algorithm. Consider, for example, output of a clustering algorithm with too many clusters compared to the number of reference speakers. In this case, all the clusters may be limited to single reference speakers and thus may have a very high purity. This, however, is not a good clustering solution because the reference speakers are split across several clusters.

To consider the effect of having too many or too few clusters compared to the number of reference speakers, the Q-Measure was introduced in (Ajmera *et al.*, 2002)[3] and is explained below:

First we define:

n_{ij}: Total number of frames in cluster i spoken by speaker j

N_s: Total number of speakers

N_c: Total number of clusters

N: Total number of frames

$n_{.j}$: Total number of frames spoken by speaker j

$n_{i.}$: Total number of frames in cluster i

[3]In (Ajmera *et al.*, 2002), this is referred to as K-Measure. In this thesis we denote it as Q-Measure to avoid ambiguity with initial number of clusters which is also denoted as K.

The purity of a cluster $p_{i.}$ is defined as:

$$p_{i.} = \sum_{j=1}^{N_s} n_{ij}^2 / n_{i.}^2 \tag{5.7}$$

The average cluster purity acp is then calculated as:

$$acp = \frac{1}{N} \sum_{i=1}^{N_c} p_{i.} . n_{i.} \tag{5.8}$$

Similarly, we calculate the speaker purity $p_{.j}$ and average speaker purity asp as:

$$p_{.j} = \sum_{i=1}^{N_c} n_{ij}^2 / n_{.j}^2 \tag{5.9}$$

$$asp = \frac{1}{N} \sum_{j=1}^{N_s} p_{.j} . n_{.j} \tag{5.10}$$

The asp gives a measure of how well a speaker is limited to only one cluster, and the acp gives a measure of how well a cluster is limited to only one speaker. The two measures are important to take into account the effect of too many or too few clusters. Finally, to get a single objective criterion, the Q-Measure is defined as:

$$Q = \sqrt{acp \cdot asp} \tag{5.11}$$

The value of Q varies between 1 (very good mapping between clusters and speakers) and 0 (very poor mapping between speakers and clusters). Note that it is also possible to define other objective measures *e.g.* arithmetic mean of asp and acp. However, the goal of the proposed evaluation metric Q (5.11) is to penalize the whole clustering performance if one of acp and asp is very poor, which is underachieved by arithmetic mean.

5.4.2 Data

The Hub4 1996 evaluation set[4] was used for most of the experiments presented in this chapter. This consists of 4, 30-minute audio files, namely, $file1, file2, file3$ and $file4$. These files have 7, 13, 15 and 20 speakers respectively. Along with speech data, the files also have significant amount of non-speech data. Although the non-speech data should be ideally removed prior to the speaker clustering step, we ran our algorithm without any pre-processing to understand the behavior of the algorithm in presence of non-speech regions as well.

5.5 System Settings and Parameters

From Sections 5.2 and 5.3 we see that the proposed clustering algorithm provides a fully automatic framework to resolve the three issues related to clustering mentioned in Section 5.2.4. However, there are a few basic system settings in this algorithm, namely the initial number of clusters (K), the initial number of Gaussian components in each cluster (M), the minimum duration (MD) constraint imposed, the type of feature vectors used and the type of initialization used to create the clusters. In the following section, we present the experimental studies in which we explore the effect of varying these settings on the system performance.

5.5.1 Feature Vectors

The feature vector to be used in this algorithm is not really a system setting, as the algorithm is designed to cluster any given set of feature vectors, with minimal human interaction. The resulting performance across different feature vectors is only limited by the information carried, and their behavior in speaker clustering paradigm. However, in the following study, we compared the performance of the most commonly used feature vectors in speaker recognition research, to find the best feature vectors among these and to be used in rest of the experiments in this chapter.

As mentioned in Chapter 2, the most common characteristic feature vectors for speaker recognition and related applications are LPCC and MFCC, while some researchers use PLPC features claiming its efficiency over LPCC features, especially in the event of acoustic mismatch (Vuuren, 1996). LPCC features were introduced for speaker verification and recognition applications by Atal

[4] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC97S66

Data	MFCC	LPCC	PLPC
File1	0.92	0.90	0.90
File2	0.88	0.83	0.87
File3	0.83	0.83	0.80
File4	0.73	0.70	0.58

Table 5.1. Comparison of performance of MFCC, LPCC and PLPC coefficients in terms of Q-measure when used as feature vectors in the proposed speaker clustering algorithm. All these feature vectors are based on spectral information derived from a short time-windowed (30ms in our case) segment of speech. 12 cepstral coefficients were extracted every 10ms in each case. The MFCC model order is fixed at 24 mel-spaced filters while 12 order all-pole model was used for derivation of LPCC and PLPC. Although the performance of all the features is reasonable, the table clearly shows the superiority of MFCC feature vectors over these other features for the purpose of speaker clustering.

(1974) and are still used (Misra *et al.*, 2003). However, more and more systems are now using MFCC features. For example, most of the participating systems in NIST speaker recognition evaluations in 1998 used MFCC features and some systems used LPCC features (Martyn *et al.*, 2000). However, no clear task-independent conclusions about their comparative performance have been made in the literature. Theoretically, both features are based on the smoothed spectral information derived from a short time windowed segment of speech and are expected to provide similar performance. The comparative performance of MFCC, LPCC and PLPC features is summarized in Table 5.1.

All the feature vectors mentioned in Table 5.1 are derived from spectral analysis of a short, 30ms time-windowed segments of speech. 12 cepstral coefficients were derived in each case, every 10ms. The model order for the MFCC features was fixed at 24 mel-spaced filters, while 12 order all-pole model was used for the derivation of LPCC features. Although the performance LPCC and PLPC features is reasonable, Table 5.1 clearly shows the superiority of MFCC features over LPCC and PLPC features. The zeroth cepstral coefficient C_0 was discarded in all the cases for the purpose of energy normalization. Moreover, although first and second order derivatives are widely used in speaker recognition research, this resulted in inferior performance in our framework. Based on this, MFCC features, without C_0 or derivatives, are used for rest of the experiments reported in this chapter.

5.5.2 Initialization

As mentioned earlier in Section 5.2.1, in the absence of any *a priori* information about the speakers and their number, the algorithm starts with over-segmentation of the data in terms of K clusters, where K is of the order of twice the number of expected number of speakers. Further discussion about the choice of this parameter K and its effect on the system performance is presented in Section 5.5.3. In this thesis, two different initialization schemes were explored. The first one is K-Means clustering algorithm (MacQueen, 1967) and second one is uniform initialization.

The K-means algorithm is a simple non-parametric clustering method and was first described in (MacQueen, 1967). This algorithm iteratively assigns vectors to K classes according to a distance metric. The K-Means clustering algorithm finds K distinct centroids in the feature space. The distance metric used for this purpose is the Euclidean distance. These K means are then deviated depending on the corresponding standard deviation to generate initializing parameters for M different Gaussian components of the GMM in each of the K clusters. However, since the individual feature vectors characterize phonetic events rather than speakers, the K centroids in the feature space are expected to point to distinct phonetic events. Thus, an algorithm using this initialization is expected to segment the data in terms of phonetic events rather than speakers. However, the large minimum duration (of the order of 200 frames) imposed in the algorithm not only prevents the data from being segmented in terms of phonetic events but also deviates these phonetic K centroids to speaker centroids from the first iteration to the second iteration when the cluster models are re-estimated based on the segmentation resulting from this initialization.

Another alternative to K-Means initialization is uniform initialization *i.e.* if there are K clusters and the length of the audio stream is L, initializing the system by assigning uniform segments of length $\frac{L}{K}$ to each cluster in the beginning. The parameters of the cluster models then can be trained on these uniform segments. Table 5.2 contains the performance of the two initialization schemes for Hub4 1996 evaluation dataset. This table shows that uniform initialization is a better choice compared to K-means algorithm in the current framework. In our analysis, it was observed that the K-means algorithm results in poor performance when all of the K clusters at the start are not occupied, meaning that in effect the algorithm starts with less (than K) number of clusters. Thus, the desired over-segmentation does not take place. This is not a problem with uniform initialization where each cluster is assigned a set of data which is then used to initialize the parameters of that cluster. This uniform initialization is used for all the remaining experiments in this chapter.

Data	K-Means	Uniform Initialization
file1	0.86	0.92
file2	0.84	0.88
file3	0.84	0.83
file4	0.61	0.73

Table 5.2. Comparative clustering performance in terms of Q-Measure for K-Means and Uniform initialization. Overall, the uniform initialization is a better choice compared to K-means algorithm.

5.5.3 Initial number of clusters (K)

The proposed speaker clustering algorithm starts with K clusters, assuming that K is much larger than the expected number of speaker in the data. In our case, K also determines the total number of parameters in the system which is kept constant throughout the algorithm. We observed in our experiments that the algorithm is not very sensitive to the exact value of this hyper-parameter, as long as the value is of the order of twice the expected number of speakers. Empirically, it was observed that assuming one speaker every 2 minutes of audio data is a reasonable choice for this parameter. This "rule-of-thumb" is followed in most of our experiments including participation in NIST evaluations presented in Section 5.7. This assumption fails when there are too many speakers in the data. however, it should be noted that in these situations, some of the speakers speak for very short durations (*e.g.* very brief interviews) and there is not enough data to train good statistical models for these speakers. As expected, it was observed that in these situations, the speaker clustering algorithm tries to make "pure" clusters for dominant speakers. Figure 5.4 shows the performance of the clustering algorithm with varying initial number of clusters. This figure shows that the algorithm is not very sensitive to the choice of this parameter as long as a sufficiently large value of K is chosen.

Furthermore, with the help of a "cheating experiment" (i.e. one that relied on ground-truth information not available in a 'real' test), we also observed that this approach (starting with over-segmentation and then merging appropriate pair of clusters iteratively) is in fact better than starting with the number of clusters equal to the known number of speakers (N_s). In this case, the algorithm was started with $N_s + 1$ number of clusters, allowing an extra cluster for non-speech data. The results of this study are summarized in Table 5.3. Thus, even if we have *a priori* information about the number of speakers in the data, it is better to exploit it in an agglomerative clustering framework rather than starting with those many states in the HMM.

Data	$K = N_s + 1$	$K = 30$
file1	0.67	0.92
file2	0.75	0.88
file3	0.61	0.83
file4	0.50	0.73

Table 5.3. Speaker clustering performance in terms of Q-Measure of the cases when the number of speakers were known and unknown. When the number of speakers (N_s) was known, the algorithm starts with $N_s + 1$ number of initial clusters. When the number of speakers was not known, the algorithm was started with over-segmentation ($K = 30$).

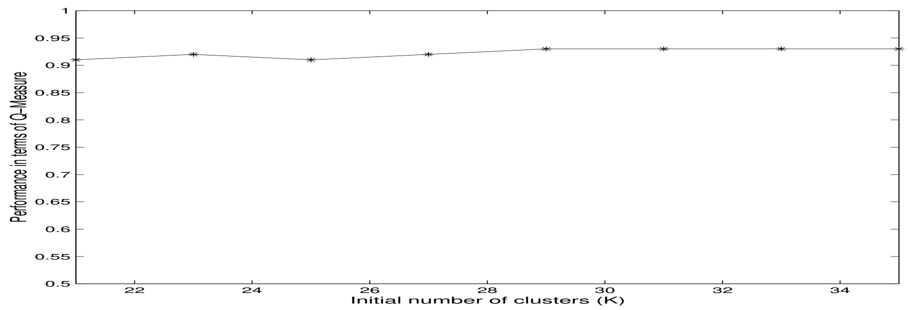

Figure 5.4. Performance of the speaker clustering algorithm in terms of Q-Measure with varying number of initial clusters (from $K = 20$ to $K = 35$). This figure shows that the algorithm is not sensitive to the choice of this parameter, as long as the algorithm is started with sufficiently large number of clusters.

5.5.4 Initial number of Gaussians per cluster (M)

As mentioned earlier, the PDFs of all the clusters in the proposed algorithm are modeled by GMMs. At the start of the algorithm, the GMM for every cluster is composed of equal number (M) of Gaussian components. Thus, if there are K clusters to begin with, there are $K * M$ Gaussians components in the whole system, and this is kept constant throughout all clustering iterations as explained earlier. This way, M is another determining factor for the total number of parameters in the system and in the following we study the effect of this hyper-parameter on the system performance.

To understand how the choice of this parameter may affect clustering performance, let us revisit the merging and stopping criterion (5.5). In Section 5.3.3, we noted that when we merge appropriate pair of clusters (clusters having data from the same speaker), we expect the term on the left hand side of (5.5) to be greater than the term on the right hand side for the merging criterion to make correct decision. Following the reasoning presented in Chapter 4 for single Gaussian density case, we note that when we try to model the similar (or same in extreme case) PDFs associated with these appropriate clusters with more parameters (equal to the sum of the number of parameters used to model the individual PDFs of these clusters), the PDF of the total data is modeled better,

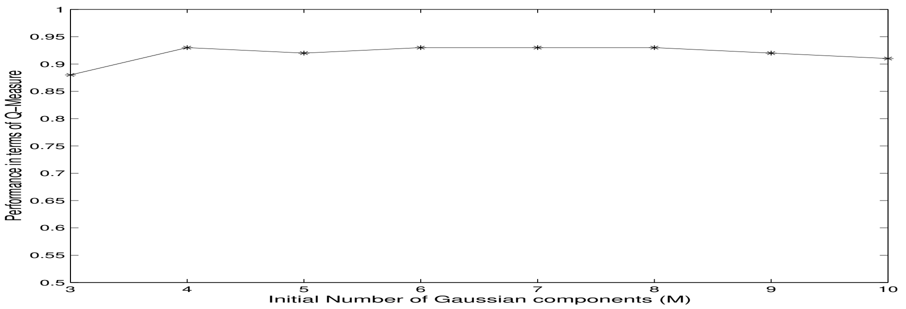

Figure 5.5. Performance of the speaker clustering algorithm with different number of initial number of Gaussian components per cluster (M). The figure shows that the performance of the clustering algorithm is quite stable with variation in the value of M.

resulting into higher likelihood of the hypothetical cluster (the term on the left hand side of (5.5)). This is even more true when the individual PDFs of the two clusters are not sufficiently modeled *i.e.* modeled using insufficient parameters (or lower value of M). Thus, the higher the value of M, the lower would be the increase in this resulting likelihood and less likely it is that the two clusters are merged. When the individual models of the two clusters are optimal, even using more parameters in the hypothetical cluster would not yield higher likelihood of the data, resulting in a no-merging decision.

However, we note that a good speaker cluster model generally requires a very high number of Gaussian components (Reynolds *et al.*, 2000) to capture all the variability of the speaker efficiently. In our framework, we use a much smaller number of Gaussian components to start with (of the order of 5) and thus the speaker cluster models are not modeled optimally. In a way, our merging criterion is based on this insufficient or non-optimal modeling of the individual cluster models and actually exploits this fact, so that the hypothetical cluster created by merging these two clusters can yield a higher likelihood of the data.

Figure 5.5 shows the effect of this parameter on the system performance on $File1$ for values of M varying between 3 and 10. We notice from this figure that the exact value of this hyper-parameter does not make significant difference in the system performance. However, following the discussion earlier, when a large value of $M = 50$ is used, very few pairs of clusters are merged, and the system performance drops down to $Q = 0.42$.

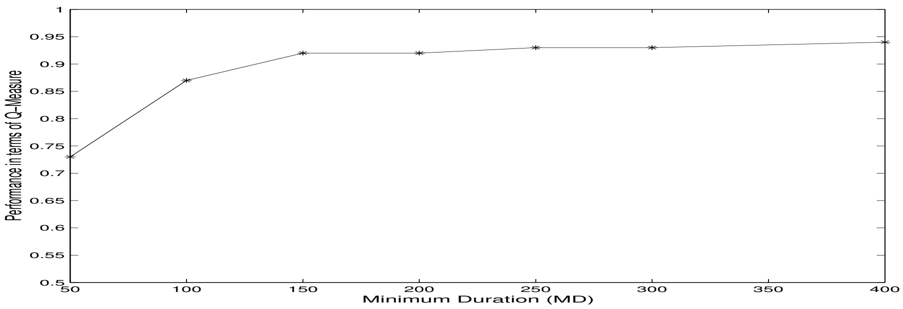

Figure 5.6. This figure shows the behavior of the proposed speaker clustering algorithm with different values of minimum duration (MD). The figure demonstrates that although a sufficiently large minimum duration (of the order of 2 seconds or 200 frames) is required, the system performance does not vary significantly with variations around this value. On the other hand, a larger minimum duration may be beneficial in some cases (*e.g.* this particular example) but in absence of any *a priori* information, this may also prove to be a erroneous as all the speech segments shorter than this duration will be misclassified.

5.5.5 Minimum Duration (MD)

As shown in Figure 5.2, each of the cluster states in the HMM is composed of several sub-states to impose minimum duration constraints. This constraint plays a major role in our system and specifically in the segmentation step where all the speaker segments shorter than this minimum duration are not considered by the segmentation step explained in Section 5.2.2. Imposing a minimum duration constraint of this form in HMM can also be regarded as providing the HMM with a prior information that speakers, especially in broadcast news speech, speak for at least a certain time to exchange useful information. In our experiments, we observed that the algorithm is not very sensitive to the exact value of this parameters as long as it is large enough (more than 150 frames, corresponding to 1.5 seconds). While an overly large MD may be beneficial in some cases, in the absence of any *a priori* information, this may also prove to be erroneous as all the speech segments shorter than this duration will be misclassified. However, if there is more information available about the minimum duration of speakers, this can be used to achieve better clustering. This is shown in Figure 5.6, where different minimum durations (from 50 frames to 400 frames) are investigated. This figure shows that the system performance is not optimal for minimum duration less than 150 frames, but is quite stable for larger value of this hyper-parameter. The use of objective function like Viterbi score was also investigated for the purpose of selection of this hyper-parameter, but we observed (as expected) that the Viterbi score is a monotonically decreasing function of this increasing MD.

Data	N_s	N_c	asp	acp	Q
$file1$	7	18	0.90	0.94	0.92
$file2$	13	14	0.92	0.85	0.88
$file3$	15	17	0.90	0.76	0.83
$file4$	20	22	0.75	0.71	0.73

Table 5.4. Speaker clustering performance in terms of average speaker purity (asp), average cluster purity (acp) and Q-Measure on four half hour broadcast news shows. N_s denotes the true number of speakers in the data. N_c denotes the number of clusters found by our stopping criterion.

5.6 Experiments and Analysis

This section presents the experiments on the Hub4 1996 dataset explained in Section 5.4.2, results and a discussion of these results. For the experiments presented in this section, the system settings were selected following the analysis presented in previous section: $K = 30$, $M = 5$, $MD = 2$ seconds and uniform initialization. Table 4.1 presents the results of the speaker clustering performance in terms of average speaker purity (asp), average cluster purity (acp) and Q-Measure. This table also shows the "optimal" (optimal according to the maximum Viterbi score objective function) number of clusters (N_c) found by our proposed automatic stopping criterion.

5.6.1 Discussion

- The number of clusters found by the algorithm N_c often corresponds to the actual number of reference speakers plus a few extra clusters for non-speech data. Although this is not true for $File1$ ($N_s = 7$, $N_c = 18$), the high value of $asp = 0.9$ and $acp = 0.94$ shows that most of the speakers are limited to their corresponding clusters and that the extra clusters are occupied with non-speech data.

- The non-speech clusters are not merged using proposed criterion (5.5). In $File1$, 33% of the data is non-speech and the algorithm results in having 11 extra clusters for these non-speech clusters. This can be explained by the fact that different kind of non-speech categories (*e.g.* clapping, music, cheering etc.) exhibit different acoustic properties. The amount of non-speech data in $File2$, $File3$ and $File4$ is 2%, 9% and 11%, respectively.

- The performance of the algorithm degrades with increasing number of reference speakers. This can be explained in two ways: First, the amount of data available for each speaker decreases, making it difficult to model some of the speakers. Second, the possibility of overlap in the feature space increases as the number of speakers increases. Moreover, with increasing number of speakers, the stated rule-of-thumb (starting from clusters equal to the duration of audio stream in minutes) may not apply. This is evident in case of $File4$ where the number of speakers is 20 and the initial number of clusters (30) can not over-segment the data. Another experiment on $File4$ with 40 initial number of clusters resulted in a much improved performance ($Q = 0.79$ compared to 0.73) and showed the importance of over-segmentation.

- The results show that on an average the average speaker purities for these dataset are very high, indicating that speakers are mostly limited to their respective clusters. This feature is very important for speaker adaptation where the emphasis is more on the average speaker purity *i.e.* acoustically similar speakers can be put in one cluster (low acp) but the speakers should not be split across several clusters asp.

5.6.2 Comparison of the proposed criterion with LLR

In Section 5.3, it was mentioned that the LLR can also be used as a merging criterion in a speaker clustering framework. For the stopping criterion using LLR, we discussed the use of thresholding (Section 5.3.1) and also proposed a framework in which this is done without the use of any threshold (Section 5.3.2). The threshold value for these experiments was estimated by K-fold cross validation of the optimal threshold values for the four datasets and the value resulting in total best performance was picked to be the threshold value for all the datasets. In this section, we study the performance of these two cases and compare them with the performance with proposed merging and stopping criteria. Table 5.5 summarizes the results obtained from these experiments.

Table 5.5 shows that while using LLR, the proposed stopping criterion (stopping the algorithm at the point of decreasing Viterbi score) not only eliminates the need for thresholding, but also results in an improved performance on an average. However, the proposed merging and stopping criterion (5.5) outperforms, on an average, the use of the LLR as a distance metric in both the proposed (Section 5.3.2) and thresholding framework (Section 5.3.1). The proposed framework has the additional obvious benefit of being a threshold and heuristic-free framework, and that no additional development data is required to tune or decide parameters.

Data	Thresholded LLR	LLR and proposed stopping criterion	Proposed merging and stopping criterion
$file1$	0.88	0.86	0.92
$file2$	0.81	0.85	0.88
$file3$	0.78	0.79	0.83
$file4$	0.71	0.75	0.73

Table 5.5. The comparative performance in terms of the Q-Measure of three systems differing in the employed merging and stopping criteria. The second column shows the results when LLR was used as merging criterion and, a threshold (empirically chosen) was used to decide the stopping point. The third column shows the results when LLR was used as merging criterion but used with a proposed stopping criterion (Section 5.3.2). The last column shows the results when the proposed merging and stopping criteria were used. The results show that the proposed stopping criterion while using LLR as distance metric eliminates the need for thresholding while also resulting in better performance on an average. Moreover, on an average the proposed merging and stopping criterion outperforms the use of LLR, both in proposed as well as thresholding framework.

5.7 Participation in NIST evaluations

In March-April 2003, NIST conducted a "who spoke when" speaker diarization evaluation for broadcast news speech[5]. This was done in the framework of Rich Transcription EARS[6] project funded by DARPA. The goal of this project is to make transcripts generated by ASR systems more readable and provide extra value to both human and machine users of the data.

Specifically, the task of speaker diarization is to identify segments of speech (speech/non-speech segmentation) and group them by speaker. This is very similar to the speaker clustering task handled in this thesis, and we used the algorithm presented in this chapter to participate in this evaluation. This section explains the evaluation metric used, data on which evaluations were run and finally compares the performance of the proposed system with other participating systems.

5.7.1 Evaluation Criterion

The evaluation criterion used in NIST evaluations is a time-based score (diarization error) which corresponds to speaker time not attributed correctly to a reference speaker. Thus, a score of 0.0 would represent a perfect segmentation. Also, it is possible to have an error > 100.0 because of the inclusion of false alarms. This error is calculated as:

[5] http://www.nist.gov/speech/tests/rt/rt2003/spring/
[6] Effective Affordable Reusable Speech-to-text

$$DiarizationError = \frac{\sum_{s=1}^{S} \{dur(s) * (max(N_{ref}(s), N_{sys}(s)) - N_{correct}(s))\}}{\sum_{s=1}^{S} \{dur(s) * N_{ref}(s)\}} \tag{5.12}$$

where the speech data is divided into S contiguous segments whose boundaries are defined by all speaker change points (including both reference and hypothesized speakers) and where, for each segment s:

$dur(s)$ = the duration of s

$N_{ref}(s)$ = the number of reference speakers speaking in s

$N_{sys}(s)$ = the number of system speakers speaking in s

$N_{correct}(seg)$ = the number of reference speakers speaking in s for whom their matching (mapped) system speakers are also speaking in s.

This error has three components; missed speaker time, false alarm speaker time, and speaker error time. Missed speaker time is the total time when there is a reference speaker speaking but the system finds no speaker or "non-speech" region. This error comes from the speech/non-speech pre-processing module, which may classify some of the genuine speaker segments as non-speech and these regions are then omitted from the final output. False alarm speaker time is the total time when there is no reference speaker speaking but system commits an error by assigning a speaker to these regions. This error also comes from the non-speech segmentation module which may keep some of the non-speech regions in the system output classifying them as speech.

It should be noted that non-speech segments were not part of the reference segmentation, *i.e.* they were omitted from evaluation. However, the task involved processing of complete audio stream. Thus, a perfect speech/non-speech segmentation module will lower the first two types of errors to zero. We applied the speech/non-speech segmentation system presented in Chapter 3 for this purpose.

Finally, speaker error time is the time when both reference and system claim speech activity, however, the system assigns a wrong speaker label to a segment. This constitutes the major part of the error especially for broadcast news speech. This is done by finding the most suited match between every cluster found in system output and the reference speakers. The speaker clustering system presented in this chapter exactly takes care of this task and only tries to minimize this last component of the error.

This evaluation metric (5.12) is developed for speaker diarization or "who spoke when" task, where the goal is to make a perfect system cluster for every reference speaker. Any time instant which deviates from this definition of the task is counted as an error in this metric. This is different from the way, these errors are handled in the Q metric (5.11) proposed and used earlier in this chapter. The Q metric penalizes every erroneous mapping between a reference speaker and a system cluster from a "clustering" point of view. Consider, for example two cases when a reference speaker is assigned to two and three clusters, respectively. The diarization error (5.12) will be the same in these two cases. Conversely, the proposed Q metric will be higher for the case when the reference speaker is split in two clusters, which is indeed a better clustering compared to the case when the reference speaker is split in three different clusters. Similarly, this difference between the two evaluation criteria is also evident for the cases when a system cluster has data from more than one speaker. Thus, we note that the proposed Q metric (5.11) can be used more generally to compare two clustering solutions, opposed to diarization error which is specific to speaker diarization or "who spoke when" task.

5.7.2 Data

All the datasets for these evaluations were released by NIST and consisted of several hours of English broadcast news. As reported in (Ajmera and Wooters, 2003), these datasets were called *dryrun* and *evaldata*. The *dryrun* data consisted of 6, 10-minute excerpts of American English broadcast news and *evaldata* consisted of 3, 30-minute excerpts of American English news shows. For these experiments as well, MFCC vectors were extracted every 10ms to be used in the speaker clustering algorithm. As an indication of the number of speakers, there were 47 male and 12 female speakers in the *evaldata*.

5.7.3 System Parameters

Based on the observations from our Hub4 experiments (explained in previous section), the system parameters were decided to fit the data sizes. For *dryrun* 10 minute datasets, $K = 15$ and $M = 5$

System	% Diarization Error
A	28.75
Proposed	**30.90**
B	32.02
C	35.05
D	58.82
E	63.59

Table 5.6. The performance of different speaker clustering systems participated in Dryrun experiments as part of RT-03 evaluations in the framework of DARPA-EARS program. In this evaluation, the proposed criterion performs better than most of the participating systems.

were used. For the *evaldata* 30 minutes datasets, $K = 40$ and $M = 5$ were used. The minimum duration of 2 seconds (which is equivalent to 200, 10ms frames) was used for both the shows. Uniform initialization was used for all the experiments.

For the speech/non-speech segmentation task, we applied the system presented in Chapter 3. However, we realized that the system presented in Chapter 3 is based on a British English ASR system, and the data in this evaluation was American English speech. From our preliminary experiments, we observed that although this system improves the overall performance, it was not efficient enough to be used as a pre-processing module for this task. Thus, after running the speaker clustering algorithm on the complete dataset, non-speech segments identified by our system were omitted from the output segmentation. The analysis of this processing is presented in next section.

5.7.4 Comparison with other approaches

For these dryrun and final evaluation sessions, 7 and 8 different systems participated, respectively, including the proposed system. The performance of all different systems participated in these evaluations are presented in Tables 5.6 and 5.7.

Table 5.6 presents the performances of the different systems participated in the Dryrun experiments involving *dryrun* dataset. The table shows that the proposed system performs better than most of the participating state-of-the-art approaches.

Table 5.7 shows the performance of the systems participated in the final evaluations involving *evaldata*. This table 5.7 shows that the performance of the proposed clustering algorithm is comparable to many state-of-the-art algorithms for this purpose. However, when this is combined with the fact that the proposed system required no training or tuning for these evaluations, the pro-

System	% Diarization Error
A	13.53
B	14.38
C	17.04
D	17.23
Proposed	**18.70**
E	19.37
F	24.50
G	27.65

Table 5.7. Performance of different systems participated in NIST speaker diarization evaluations. The performance of the proposed system is highly competitive with other approaches, with a clear advantage of being a threshold/adjustment free approach.

Error Type	Without SND	With SND
Missed speaker time	0.1	2.8
False Alarm Speaker Time	11.6	7.0
Speaker Error Time	9.8	8.9
Total	21.5	18.7

Table 5.8. The analysis of the errors for the cases when a speech/non-speech discrimination (SND) was applied and when it was not applied. The SND module significantly improves the overall speaker diarization performance.

posed algorithm holds a clear advantage. Moreover, during our detailed analysis, it was observed that the *evaldata* consisted of a lot of non-speech data (11.6% of the scored time) and thus, the speech/non-speech segmentation also held an important key in the performance of different systems. For clarity, we would also like to mention that the numbers presented in Table 5.7 are a result of both speech/non-speech discrimination (SND) and speaker clustering.The system presented in Chapter 3 was used for SND for this purpose. Table 5.8 presents the results of the proposed system with and without the speech/non-speech discrimination (SND).

As expected, speech/non-speech discrimination (SND) helps the system by lowering down the FA speaker time errors. However, it results in increased missed speaker time. While comparing this break-down of the results with other systems, we realized that our SND performance (reflected by false alarm speaker time and missed speaker time) was considerably worse than the other systems, resulting in overall higher diarization error. On the other hand, our speaker error time was much lower compared to other systems. The breakdown of the performance of different systems in terms of three kind of errors is shown in Figure 5.7. As mentioned earlier, we applied the system presented in Chapter 3 for this task, and this system is primarily based on a British English ASR. Whereas, the data used in these evaluations was American English broadcast news speech. On the other

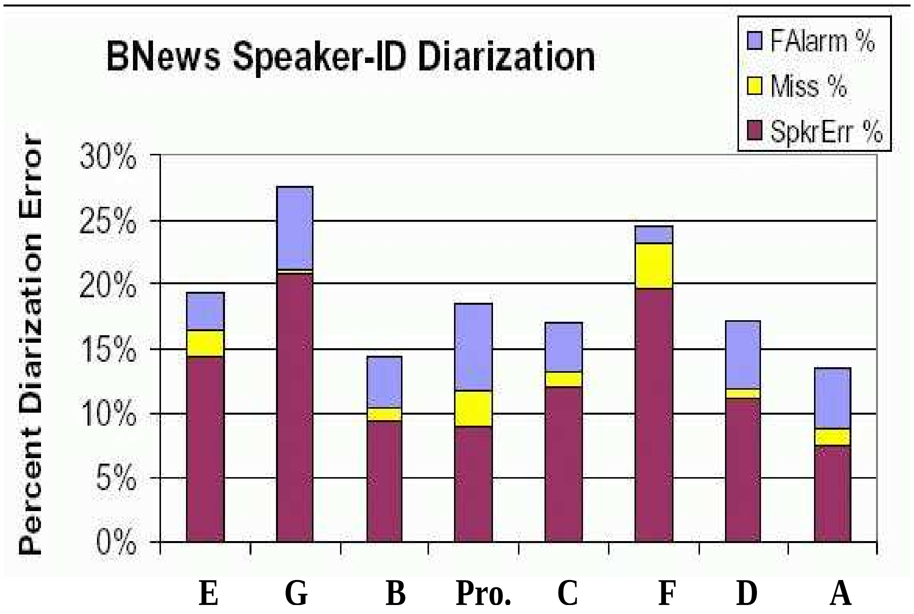

Figure 5.7. Breakdown of performance of different participating systems in RT-03 speaker diarization evaluations in terms of False Alarm Speaker Time (top partition), Missed Speaker Time (middle partition) and Speaker Error Time (lowest partition). The performance of the proposed system is labeled as Pro. The figure shows that the while the performance of the proposed system in terms of speaker error time is very competitive compared to other systems (second after 'A'), the performance in terms of other two types of errors is considerable poor.

hand, the performance of the speaker clustering algorithm alone (in terms of speaker error time) was in fact superior to most of the participating systems.

5.8 Conclusion

This chapter addressed the problem of speaker clustering where the goal is to segment the speech data and group them by speakers. A HMM based agglomerative clustering framework was developed for this purpose, where the system starts by assuming large number of speaker clusters. Following this, best pair of clusters according to a distance/similarity metric is chosen and merged. This is done iteratively until a stopping criterion is met. It was pointed out clearly that in most of the previous approaches, this stopping criterion is based on heuristics or empirically chosen threshold values.

This chapter presented a novel distance metric, which can be used as a distance metric for merging two clusters. The distance metric has the form of LLR where the number of free parameters in the two models is forced to be equal. This makes the important difference that unlike LLR which is always a negative value, the value of the proposed distance metric ranges from negative to positive values, making zero a natural threshold for this problem. The important property of this metric is that it does not rely on any heuristic or threshold values and the merging decision is made automatically. The clustering process is stopped at a point when no more pair of clusters satisfy this merging criterion.

Although the proposed algorithm is efficient and requires a minimum of human interaction, there are still a number of basic system settings involved in this system. A detailed analysis of the system performance with variations in these settings was carried out. It was shown that the system performance does not depend heavily on the exact choice of the values of these parameter within a reasonable range. Moreover, the performance of the proposed merging and stopping criteria was compared with that of log-likelihood ratio (LLR), and it was shown that the proposed criteria outperforms LLR in most of the situations.

Finally, details of our participation in NIST evaluations were presented where the proposed algorithm could be compared directly with other state-of-the-art approaches in this domain. The comparison showed that the proposed system performs quite comparable to other approaches on an average. This together with the fact that unlike other approaches, the proposed algorithm does not require any threshold adjustment and hence development data shows that the proposed algorithm holds a clear advantage over other state-of-the-art approaches.

Chapter 6

Applications

The problems and proposed solutions for speech/music discrimination, speaker changed detection and speaker clustering have been discussed individually in Chapters 3, 4 and 5 respectively. However, it is clear that in practical applications, these problems do not arise independently. For example, if broadcast news has to be indexed in terms of speakers, it is rarely the case that the data contains only speech segments. Moreover, depending on the application, the algorithms also need to be modified. For example, in the task of online TV captioning, only part of data is available for processing. Thus, all the modules of audio segmentation need to run in an online fashion as opposed to offline algorithms (such as speech/segmentation system in Chapter 3 and speaker clustering algorithm in Chapter 5) which make global decisions by utilizing all the data.

This chapter presents two applications that were developed as part of relevant projects during this thesis work. These applications are built using the basic modules and algorithms presented in the previous chapters, with task dependent modifications. These algorithms are appropriately modified whenever needed. The first application concerns online audio indexing, *i.e.* as the audio stream arrives, it should be first verified whether it is a speech segment or not and, in case it is speech segment, a speaker label should be assigned to the segment to identify "who is speaking". Also, the speaker changes and their labels should be found on-the-fly and consistently. This application is explained in Section 6.1.

The second application relates to the problem of meeting summarization. After the meeting has taken place, the recorded audio should be segmented and indexed so that it can be navigated easily by the content. This navigation will include finding the identities of the attendees of the meeting, identity of the person who made a presentation or all the segments spoken by a particular speaker. The application proposed in this chapter facilitates this by segmenting the audio data jointly in terms of both speaker identities and their locations. This application is explained in Section 6.2.

6.1 Online Audio Indexing

A recent research domain that has received much attention is multimedia content analysis, which concerns the automatic extraction of semantic information from multimedia documents (Wang *et al.*, 2000). This has a variety of applications, such as indexing, retrieval and browsing of archives of broadcast news programmes (THISL Project, 1997) or sports videos (ASSAVID Project, 2000). This application focuses on extraction of information from the audio stream of a multimedia document. This audio indexing system has been integrated into retrieval systems in the ASSAVID Project (2000) and CIMWOS Project (2000) within the European Information Society Technology (IST) Programme. Both of these projects investigate more general multimedia content analysis, in which the audio processing system is integrated with video processing modules such as recognition of on-screen text, face recognition, and object recognition.

A block diagram of the audio indexing system is shown in Figure 6.1. It consists of processing modules for segmentation, classification, clustering, recognition and resynchronization. These modules interact to process the input audio stream in an online manner, producing output XML annotations in real-time. The modules are described in more detail in the following sections.

6.1.1 Segmentation

The input audio stream is first processed by the segmentation module, which detects transitions in the audio characteristics. The module is expected to find change points whenever there is a transition from music to speech or from speech to music. Moreover, within speech regions, the module also detects transitions from one speaker to another. However, the task in practice is a little more complicated as these transitions are not so obvious all the times. For example, in broadcast news shows, the anchor starts speaking with a theme music in background (referred to as mixture regions as explained in Chapter 3) and the intensity of this music gradually fades. Sometimes, the

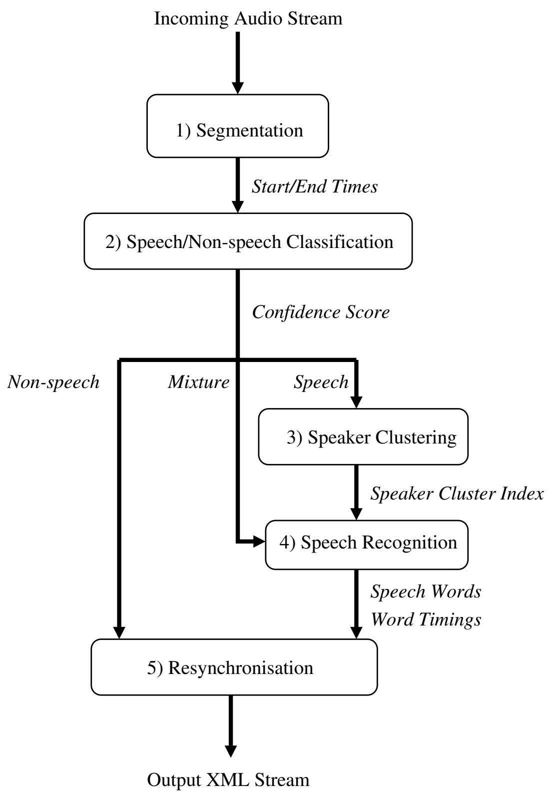

Figure 6.1. Block diagram of system. *Italicized* text indicates meta-data output after each processing module. The input audio stream is first segmented in terms of homogeneous acoustic regions using the system presented in Chapter 4. Each of these homogeneous segments is then classified as speech, music or "mixture". Two consecutive music or mixture segments are combined back to form one big segment. The speech segments, on the other hand are grouped together (clustered) in an online fashion. A newly arrived speech segment is compared with all previous speaker clusters using the merging criterion proposed in Section 5.3.3. If this segment does not match with any of the previous clusters, a new speaker cluster (identity) is created. An ASR system is then employed to convert all speech and mixture regions into text. Finally, all this meta-data information is properly time-tagged and written in the form of an XML output.

environmental sounds change while a reporter is speaking. Thus, many times it is not obvious even to a human listener, whether a change point should occur or not. These situations are handled by the segmentation module in conjunction with the following speech/non-speech discrimination block.

The algorithm and online framework presented in Chapter 4 is used for the segmentation module. We note that this technique finds speaker changes in a given audio stream assuming that the input is composed of all speech segments. However, in the complex situations as mentioned above, the algorithm is forced to find change points whenever there is a remarkable change in the acoustic properties of the input signal. For example, the algorithm will find a change point in the above mentioned example of an anchor, when the background switches from background music to clean speech.

The behavior of this module is quite obvious during speech and mixture regions as the algorithm tries to find speaker turn points or points where the background acoustics within a single speaker turn changes significantly. However, during music regions, the acoustic properties of the signal are dynamically changing and hence this module is not expected to find boundary points at the beginning and end of these music regions. In these situations, the algorithm in fact finds multiple change points. This problem of finding too many segments during music is resolved in conjunction with the following speech/non-speech discrimination block where all consecutive music segments are identified and combined together to form single music segment as described in the following section.

6.1.2 Speech/Non-speech Discrimination

Once a homogeneous audio segment has been detected by segmentation module explained in the previous block, the next task is to classify it as speech or non-speech. We refer to this process as speech/music discrimination as opposed to speech/music segmentation where the task also includes finding segment boundaries.

The techniques explained in Chapter 3 are used for this module. The system explained in Chapter 3 is primarily designed for speech/music (as opposed to speech/non-speech) segmentation, however, one of the conclusions made about this system was that it essentially tries to classify the audio signal in terms of "recognizable" and "non-recognizable" segment. This is further verified in this application as the data used for experiments (broadcast news speech) in this application has music as the dominant non-speech category but is also composed of several other non-speech sounds. Moreover, our use of system explained in Chapter 3 is that it is primarily based on functioning of an ASR system which is also an integral part of an audio indexing application.

First, PLPC feature vectors extracted from each homogeneous segment found by the segmentation module are presented at the input of an MLP which is trained to recognize clean speech. This results into a set of posterior probabilities every time frame. A sequence of two dimensional feature vector composed of entropy (3.2) and dynamism (3.4) is extracted from these posterior probabilities as explained in Chapter 3. This feature vector sequence is then presented to a secondary MLP, which outputs the posterior probability of speech or music classes given input feature vector sequence. Arithmetic mean of these probabilities over the whole segment (homogeneous segment detected by the segmentation module) serves as a confidence score as given by (3.5). The confidence score is also explained in detail in Chapter 3.

Depending on the confidence score, a segment is identified as speech, music or mixture signal. If the confidence score lies below a lower threshold, the segment is classified as non-speech or music. On the other hand, if the confidence score is above an upper threshold, the segment is classified as clean speech. Segments having scores between these two thresholds are marked as mixtures *i.e.* having speech with a background non-speech/music element. Two consecutive music segments are combined to form a single music segment, thereby also solving the problem mentioned in previous section of over-segmenting the music regions. Two consecutive mixture segments are combined as well and no speaker identity is given to these segments.

The identified speech segments from these blocks, however, need further identification in terms of speaker identity. This is done in the speaker clustering step, as explained below.

6.1.3 Speaker Clustering

Segments that have been classified as *speech* (not including *mixture* segments) are subsequently passed through a speaker clustering module, which assigns a cluster index to the segment. Speaker clustering information is useful for improving the segmentation, and for grouping segments belonging to similar speakers across a database for browsing and retrieval. It could also potentially be used in adapting acoustic models for speech recognition, or as a pre-processing step to full speaker identification, although these are not implemented in the current system.

Whenever a new speech segment comes in, it is compared with all existing speaker cluster models. If the new segment is not found to match with any of the existing clusters, a new speaker cluster is formed. Every cluster is composed of a set of data and an associated PDF in the form of a GMM. The merging criterion (5.5) presented in Section 5.3.3 is used to decide if the latest segment belongs to any of the previous speaker clusters or not.

To further improve upon the possible errors that have been made in the segmentation step, this clustering module also merges any contiguous speech segments belonging to the same speaker cluster.

6.1.4 Speech Recognition

This module is not a focus of this thesis, however, for the completeness of the system description, the information about this module is also mentioned below in brief.

The final stage of the processing passes each *speech* and *mixture* segment as input to a large vocabulary hybrid ASR system. The MLP used in this recognition system is same as the primary MLP of speech/non-speech discrimination module and was trained in the framework of THISL Project (1997). Since the phonetic posterior probabilities for this are already calculated during speech/non-speech discrimination module, they can be directly used for this without any additional processing.

The language model for the ASR system is trained using a corpus containing text from BBC News and English newspaper and news-wire services. To help in ranking results in any eventual retrieval system, each word in the speech recognition output is assigned a confidence value. In this case, this is the speech confidence score output by the speech/non-speech classification module described above, and as such all words within a segment are assigned the same confidence.

The above modules are implemented as separate processes that run in parallel, with inter-process communication via pipes. Because it is possible for some segments to pass through the system faster than others (music and mixture segments involving less processing than speech segments), it is necessary to resynchronize the meta-data at the output. A demonstration of this application can be found at IDIAP website[1] and a snapshot of this demonstration is already shown in Figure 1.2 in Chapter 1.

[1] http://www.idiap.ch/pages/contenuTxt/Demos/demos.php

6.1.5 System Performance

For experiments and evaluating the performance of the different modules explained above, BBC broadcast news show was used as data. The total duration of this audio stream is 1510 seconds, labeled in terms of speakers and "excluded regions" which are considered "non-speech" for this evaluation. The total duration of such "excluded regions" is 360 seconds.

The evaluation metrics used for these different modules have been explained in previous chapters of this thesis. The segmentation module is evaluated in terms of precision (PRC), recall (RCL) and F-Measure explained in Chapter 4 in Section 4.5. RCL is an indication of how well the true change point are spotted by the system in a certain time window (1 second in our case) and PRC is an indication of how precisely the system finds true change points without finding other change point (false alarms).

Speech/non-speech segmentation module is evaluated in terms of percentage of total frames classified correctly, referred to as frame accuracy. This measure is also used for the purpose of evaluation in Chapter 3.

Speaker clustering module is evaluated in terms of average speaker purity (asp), average cluster purity (acp) and their geometric mean referred to as Q-measure as explained in Chapter 5 in Section 5.4.1. More specifically, asp provides an indication of how well (on an average) a particular speaker is assigned to a particular cluster, and not split across several clusters. On the other hand, acp provides an indication of how well (on an average) a particular cluster label is assigned to a particular speaker.

The results for different modules using respective evaluation metrics are summarized in Table 6.1. Our system detected total 181 seconds of "mixtures". Out of these, 87 seconds of mixtures were labeled as speech/speaker and remaining 94 seconds were labeled as "excluded region" in the ground-truth. In our analysis, we observed that these "excluded region" are mostly commercial advertisements with mainly music but also with a significant speaker activity and hence it is not surprising that our system detected most of these regions as "mixtures". The other "mixture" regions were mostly beginning and end of programme when an anchor is speaking with background music.

There are 44 true reference change points (from one speaker to another and from speech to music) in the test data. The *offline* system (as explained in Chapter 4) detects a total of 85 changes.

Out of these 39 changes correspond to the true reference change points, resulting in $RCL = 0.89$, $PRC = 0.46$, $F = 0.61$. Most of the 46 inserted changes, occurred during pure music regions. As expected, the speech/non-speech discrimination block merges most of these short consecutive music regions and hence improves the precision (PRC). The number of segments after speech/non-speech discrimination is 66. This results in improved segmentation performance of $RCL = 0.84$, $PRC = 0.55$ and $F = 0.67$.

If we consider the "mixture" regions (as identified by our system) as speech for speech/non-speech segmentation evaluation, the performance in terms of frame accuracy is 86.2%. In our analysis of errors related to this module, we observed that most of these errors are made during music regions *i.e.* music frames are classified as speech. However, as mentioned earlier, most of these identified "mixture" regions corresponded to the regions of commercial advertisements which are indeed "mixtures" but labeled as "excluded regions" and we considered them as non-speech for this evaluation. This analysis was also supported by the performance of the system presented in Chapter 3 on the same data, which provided 95% frame accuracy. If we do not consider these regions (total 93 seconds) in evaluation, the speech/non-speech classification frame accuracy is 92.4%.

The performance of the speaker clustering algorithm in terms of Q-Measure is 0.72. Note that the mixture regions are not clustered with regular speaker clusters and hence for the purpose of evaluation, a new speaker identity was assigned for every mixture segment. This resulted in $Q = 0.72$. In our analysis, we observed that this is also similar to the performance of the offline system (as presented in Chapter 5 which is $Q = 0.72$. This indicates that online clustering done in this way is a good alternative way to the previously presented agglomerative clustering framework. However, we note that the current clustering does not deal with segments classified as non-speech and mixtures, whereas the offline clustering deals with these segments as well.

Also, we note that this system was implemented in a way that only a limited number of samples are considered to characterize a cluster, and this also results in a much reduced computational complexity compared to the offline algorithm presented in Chapter 5, where the complexity of the algorithm in quadratic in the number of clusters.

Module	Metric	Performance	Offline Performance
Segmentation	Precision	0.55	0.46
	Recall	0.87	0.89
	F	0.67	0.61
Speech/Non-speech	Speech Frame Accuracy	98%	99%
	Music Frame Accuracy	49%	82%
	Total Accuracy	87%	95%
Speaker Clustering	Average Speaker Purity	0.79	0.85
	Average Cluster Purity	0.66	0.61
	Q	0.72	0.72

Table 6.1. Summary of performance for different modules of the proposed online audio indexing application. The table also presents the performance of corresponding offline system presented in one of the previous chapters. As expected, some of the errors made in the segmentation step are recovered in later stages of speech/non-speech classification and speaker clustering and hence system performance improves over the offline system. We note a big degradation in speech/music classification accuracy for the compared to the corresponding offline system, mainly because of the inclusion of the "mixture" class, which is considered as speech for the evaluation purposes. The performance of the speaker clustering done in this online fashion is similar to the performance of the offline system, however, note that the current system does not cluster non-speech and mixture segments.

6.1.6 Application to Multimedia Content Analysis

The audio annotation system described in this chapter has been integrated into two multimedia annotation systems in the scope of the ASSAVID Project (2000) and CIMWOS Project (2000) European projects. The ASSAVID system is designed to address the requirements of archivers in the BBC sports library. The system architecture is detailed in (Christmas *et al.*, 2003). In the context of the ASSAVID system, the non-speech/music confidence score (converse of the speech confidence) was found to be a particularly useful cue for higher level processing. In sports video, it was found that a high value for this score generally indicated either significant crowd noise (and thus significant moments) or music (often national anthems). Similarly, the CIMWOS system contains a number of integrated audio and visual processing modules to automatically annotate general multimedia data. Specific modules include face detection and recognition, object recognition, text detection and recognition, audio segmentation and speech recognition. The XML output of these modules is merged in an indexing and retrieval system.

In such a multimedia retrieval system, particularly useful aspects of the audio system are the segment-based speech confidence score and the speaker cluster information. The correlation between the confidence score and speech recognition accuracy (Figure 3.5 in Chapter 3) can be exploited in searches to enlarge or restrict the number of hits. Along with speech words, and visual information, the speaker clustering information can be useful in retrieving similar segments from

Figure 6.2. IDIAP Smart Meeting Room

across an archive. Further work could integrate this with visual information to more accurately cluster presenters or reporters across an archive.

6.2 Clustering Speakers and Their Locations in Meetings

This application relates to the task of meeting summarization in the framework of M4 Project (2002). This project is concerned with the construction of a demonstration system to enable structuring, browsing and querying of an archive of automatically analyzed meetings, which have taken place in a room equipped with multi-modal sensors. One such "smart" meeting room (SMR) has been setup at IDIAP (Moore, 2002). Figure 6.2 shows the setup of this SMR.

The IDIAP SMR is a 8.2m $\times$3.6m $\times$2.4m rectangular room containing a centrally located 4.8m $\times$1.2m rectangular table (suitable for seating 12 people). A white-board and retractable projector screen occupy the wall at one end of the room. This room is capable of recording high quality, multi-channel, audio-visual meeting data. The current configuration uses three cameras with wide-angle lenses, six lapel microphones, and two 8-element microphone arrays to record meetings containing up to six participants. All channels of audio and video are accurately synchronized and time-stamped. The overall motivation behind such a setup is to investigate how information from meetings can be captured, stored, structured, queried, and browsed using multi-modal sensors, analysis, and user interfaces. The aim is to provide techniques that will help people quickly obtain required information from a meeting archive without having to listen and view entire recordings. This will assist both people who have missed a meeting, as well as those who attended but need to recall certain details.

The presented application concerns processing of audio signal acquired from meetings. Ideally, this should be integrated together with processing of other modalities: vision, gestures, text, touch etc. to generate richer and more accurate information set, but it was not focus of this thesis work. Specifically, in this work, we concentrate on segmenting the audio archives from these meetings in terms of speakers and their locations (Ajmera *et al.*, 2004a). Unlike speaker clustering which segments the data only in terms of speaker and unlike location based segmentation (Lathoud and McCowan, 2003) which segments the data only in terms of the location (direction) of the sound source, the goal of this application is to segment the data both in terms of speaker identities and direction of the sound source. To the best of our knowledge this has not been tried before.

This is an important step toward automatic meeting summarization as it enables the automatic answering of questions like "who is speaking and at what place?". For example, based on this knowledge, a user could query a structured database to show "the last presentation made by such person" or "the last meetings attended by such person". Conversely, a user may simply want to know who attended a meeting that he missed. Globally such structuring of meeting recordings also greatly enhances the playback experience, insofar as the user can quickly access information that is relevant to him (survey in Cutler *et al.* (2002)).

This work was done in collaboration with a colleague at IDIAP (Mr. Guilluame Lathoud) who has developed an efficient and precise technique for segmenting the audio data from meetings in terms of locations (direction) of the sound source (Lathoud and McCowan, 2003). Given that the speakers do not change their locations during a meeting or if the processing is only limited one meeting, this would also reflect in accurate speaker segmentation and clustering. However, in real-life meetings people may stand up and move e.g. to the presentation screen and switch places. Moreover, it may also be desirable to process several meeting archives at a time to access all the speech made by a particular speaker across these meetings. In these situations, although the location based segmentation would result in precise audio segments but this processing is incomplete without meaningful and consistent identities of these segments.

It was realized that this issue can be resolved in conjunction with the speaker clustering algorithm presented in Chapter 5, where the acoustic properties of the signal can be used to provide consistent labels (clustering) to the segments found by location based segmentation. This application also provided us a framework for testing the performance of the proposed clustering algorithm on meeting data, which is different from broadcast news speech data in many aspects. The broadcast news speech data is mostly single channel, with relatively long speech segments. The meeting data on the other hand are real discussions, short speech segments with frequent interruptions and overlaps. Therefore, it may be more difficult to train accurate speaker models and obtain accurate speaker segmentation in such environment.

Thus, the goal of the proposed system is to design a framework for jointly segmenting and clustering the audio stream, while utilizing information from two sources *i.e.* acoustic and directional properties of the signal.

6.2.1 Feature Extraction

Two sequences of characteristic feature vectors are extracted from the signal; acoustic and location feature vector sequences. Acoustic features are derived by short term spectral analysis of single-channel audio stream. MFCC coefficients are one example of such processing and are used for this application. Location features are derived from multi-channel cross-correlation analysis. Specifically, the microphone array in SMR is used to locate the dominant sound source at each time frame in terms of bearing: at each time frame t, an estimate of azimuth (angle in horizontal plane, θ_t) and elevation (angle in vertical plane, ϕ_t) is derived. A single source localization technique based on the SRP-PHAT measure (DiBiase *et al.*, 2001) is used for this purpose, due to its simplicity and suitability for reverberant environments.

6.2.2 Clustering Locations

One way to achieve the joint segmentation would be to concatenate the two feature vectors (acoustic and location feature vectors) and then using the clustering framework defined in Chapter 5, thereby also deriving the number of locations automatically. There are several problems with this approach.

First, each class in this framework corresponds to every possible speaker speaking at every possible locations and this results in large number of classes. A direct consequence of this is that there is not enough data for every class to make a good statistical model based on GMM, which is clearly needed for clustering algorithm proposed in Chapter 5. Second, we are looking for a partition of the two-dimensional space (θ, ϕ) into simple, connected regions. This leads to modeling with single Gaussian densities rather than GMMs. On the contrary, using the GMM framework presented in Chapter 5 would lead to non-connected clusters, *i.e.* each cluster potentially containing various non-connected regions of the space. Since we cannot use a GMM framework for clustering location feature vector sequence, we decided to use K-Means algorithm to achieve this, where the distance metric used is the angle between two feature vectors.

6.2.3 Combined System

The final goal of the combined system is to derive a joint segmentation of the audio archives in terms of speakers and their locations. We achieve this in an HMM framework similar to the one used in Chapter 5 except that the number of classes in this case is much larger, as already mentioned above. Each class in the HMM corresponds to a particular speaker speaking at a particular location. After the data is segmented in terms of these classes, the speaker "clusters" are created by getting all the data for this speaker from all possible locations and the parameters of the PDF (a GMM) of this cluster are trained using this data. This resolves the above-mentioned problem of having too many classes and little data for each of them. On the other hand, the segmentation of the data is derived by considering the information in both sets of feature vectors. However, it is not done by concatenating these two sets (for the problems mentioned above) but by assuming independence between the two information sources and accordingly factorizing the joint PDF. The resulting system, thus, consists of the following steps:

1. The K_L locations are partitioned using the K-Means algorithm, resulting in to K_L centroids and their corresponding standard deviations. From this, a Gaussian PDF is defined for each location cluster. These PDFs are unmodified in the rest of the algorithm.

2. An ergodic HMM with minimum duration constraints is constructed with $K_L \times K_S$ number of classes. In other words, we define a class for each possible active speaker (K_S) at each possible

location (K_L). A joint Viterbi segmentation of the two streams (location and acoustic features) is achieved in terms of $K_L \times K_S$ classes following independence between speaker location and speaker identity. The PDF of each state q_{sp}, q_{loc} is therefore:

$$p(X_{sp}, X_{loc}|q_{sp}, q_{loc}) = p(X_{sp}|q_{sp}) \cdot p(X_{loc}|q_{loc}) \tag{6.1}$$

where X_{sp} and $q_{sp} \in [1 \cdots K_S]$ are respectively an acoustic feature vector and a speaker state, while X_{loc} and $q_{loc} \in [1 \cdots K_L]$ are respectively a location feature vector and a location state. It should also be noted that K_S is the current number of speaker clusters and changes following a merging in step 5.

3. From the joint segmentation in terms of $K_L \times K_S$ classes, all the data belonging to each of the K_S speaker clusters is collected, and a cluster model (GMM) is re-trained using EM algorithm to maximize the log-likelihood of this data.

4. Based on these speaker clusters (characterized by a PDF and data associated with it), the cluster merging criterion (5.5) defined in Chapter 5 is applied to find the best candidates for merging.

5. If the cluster merging is possible, then these speaker clusters are merged. The model of the new cluster is re-trained and the algorithm goes back to step 2. Else, the algorithm stops and the segmentation achieved in step 2 is the output of the system.

6.2.4 Experiments and Evaluation

Data

6 four-people meetings, which are part of a corpus that is fully described in (McCowan *et al.*, 2003a)[2], were used for the experiments. Each of these meetings are 5 minutes long in duration and the concatenation of these meetings can be viewed as a single 30 minute meeting, where the participants and their places (location) change. The room setup is explained in (Moore, 2002) and is shown in Figure 6.3.

[2]These meeting recordings are publicly available at `http://mmm.idiap.ch/`

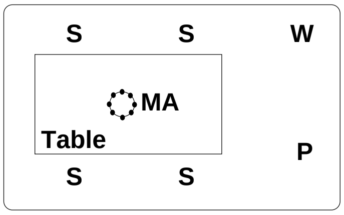

Figure 6.3. Recording setup. "S" denote participants' seats, "P" presentation screen, "W" the white-board and "MA" the microphone array.

Evaluation Metric

To evaluate the speaker segmentation performance, a precise ground-truth (GT) segmentation of the recordings was created by an independent observer. There are 6 GT speaker clusters and 1 GT silence cluster. Each GT cluster is segmented independently in terms of "activity" and "non-activity". In other words:

- In the case of a GT speaker cluster, "active frame" means this person is speaking - which does not exclude other speakers also being active. Indeed, the data does feature overlaps.

- In the case of the GT silence cluster, "active frame" means nobody is speaking.

As explained in Section 6.2.3, our scheme produces a single speaker segmentation, therefore not allowing overlaps between speakers; whereas the GT does include overlaps between speakers. In order to compare with the GT, we transformed our single segmentation into a series of active/non-active segmentations, one for each speaker cluster. The data contained very short speech segments, 50% of them being shorter than 960 ms. Therefore we could not use the usual segmentation measures of precision and recall. Two metrics were used: Frame Accuracy (FA) and Half-Total Error Rate (HTER). FA is the overall proportion of correctly classified frames. HTER is the average of False Alarm Rate (FAR) and False Rejection Rate (FRR). FAR is the proportion of erroneous frames in the active frames of the result. FRR is the proportion of erroneous frames in the non-active frames of the result. We used HTER to determine the best combinatorial match between GT speaker identities and result speaker identities.

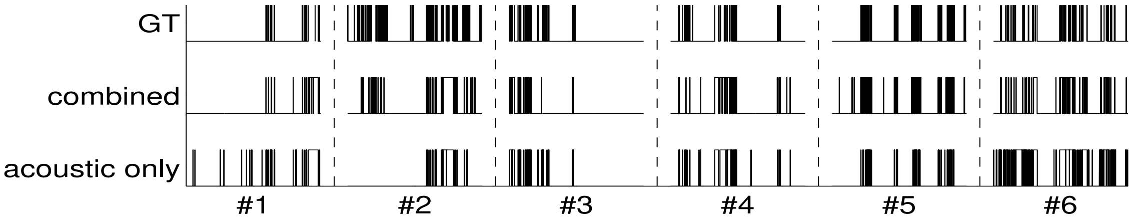

Figure 6.4. Time-line of speech activity for the 6 speakers. "GT" row shows the ground-truth, while the "combined" and "acoustic-only" rows show the respective results from the two systems. Each column spans over 1800 seconds of meetings.

Results

We first ran the acoustic clustering algorithm described in Chapter 5 alone, then the combined system. While both systems provided the correct number of acoustic clusters (6 speakers and 1 silence), the quality of the speaker segmentation was improved in terms of both HTER and FA, by using the combined system, as shown in Table 6.2.

System	HTER	FA
Acoustic Only clustering	19.2	92.6
Combined Clustering	17.3	94.6

Table 6.2. HTER and Frame Accuracy (FA) in percentages for the acoustic only and for the combined system. The results show that the combined system improves over the acoustic clustering.

The actual speaker segmentation results and ground-truth are shown in Fig. 6.4. It shows the entire speech/silence time-line for each speaker. This high-level view shows the concatenation of the 6 meetings, in other words it shows how usable the results are to answer the question "who attended that meeting?". Improvement brought by the combined system is clearly visible for speakers #1 and #2, while results for the other speakers are similar to the acoustic-only results.

On the other hand, the performance of combined clustering in terms of locations is very similar to the performance of the locations based clustering alone. In fact, the combined system makes use of the precise location based segmentation and uses the acoustic cues to provide consistent labels to these segments.

6.3 Conclusions

This chapter presented two applications which are based on the building blocks from previous chapters of this thesis. The first application concerns online audio indexing which is important for many real-life applications like online TV captioning or broadcast news speech summarization. This application consists of an integrated system of audio processing tools to extract meta-data (information other than content of the data) from an audio document. The system accepts an input audio stream, partitions it into homogeneous segments, and classifies these segments as being either clean speech, noisy speech (mixture), or non-speech. The clean speech segments are additionally assigned a speaker label, and then words in the speech and mixture segments are recognized. All time-stamped meta-data is written to an output XML stream. This XML output can then form the basis of a system for browsing, indexing and retrieving of multimedia documents. The system runs online, in real-time operation on a modern PC.

The second application concerns automatic analysis of audio stream for structuring, browsing and querying an archives of meetings. The archived meetings are held in a room equipped with multi-modal sensors (smart meeting room). The goal of this audio processing was to segment the audio stream jointly in terms of speaker identity as well as the direction of the sound source. An HMM framework was adopted for this joint segmentation. The information from two streams (acoustic and location) is effectively combined during segmentation by assuming an independence assumption between the two and factorizing the PDFs. However, the clustering of the two feature sequences is carried out independently, also resolving the problem of having too many classes and little data. Our experiments on 6 concatenated, 5-minute meetings (corresponding to a single 30 minute meeting) data recorded at IDIAP SMR showed that the proposed system not only effectively segment the data jointly in terms of “who spoke when and where?” but also the speaker clustering (consistent labeling) performance is improved compared to the clustering done using only acoustic information in the audio. This clearly shows that the precise location segmentation is very helpful for speaker segmentation and should be used in scenarios like smart meeting room where microphone array is part of the setup.

Chapter 7

Conclusion

The major contributions of this thesis can be broken down into three parts related to three different tasks or stages of audio segmentation, namely speech/music segmentation, speaker change detection and speaker clustering. While the specific contributions remain module dependent, the underlying common theme has been to develop robust (threshold or heuristic free) and unsupervised techniques for audio segmentation. Following sections present brief summary and specific contributions related to different tasks.

7.1 Speech/Music Segmentation

The proposed technique for speech/music segmentation (Chapter 3) is primarily based on the functioning of a hybrid ASR system (Bourlard and Morgan, 1994), with an MLP estimating the posterior probabilities of the phonemes in the language being recognized. The features (entropy and dynamism) used in this work are derived from the analysis of these posterior probabilities and were introduced for the purpose of discriminating between speech and music classes by (Williams and Ellis, 1999). In the present work, we further studied these features in detail and used them in a HMM framework, where both GMM and MLP experts were investigated for the purpose of estimating state emission probabilities. This framework was used to segment continuous audio streams in terms of speech and music classes. The system was tested on broadcast news data which exhibits real-life scenarios (different speech and music styles, different durations, overlapping speech and music, etc.). We make the following conclusions from this work:

- Entropy and dynamism features have complementary information and make a strong discriminatory feature set (better than MFCC) for speech/music segmentation.
- There is no remarkable difference between the performance of GMM and MLP experts for this task. However, GMM parameters can be trained in an unsupervised way without compromising on performance.
- Since the output of the MLP expert are real probabilities of speech and music classes, defining a meaningful confidence score over a segment which can indicate proportion of speech in that segment is easier. This measure can be used to provide metadata information in terms of *mixtures* (defined in this thesis as regions where both speech and music signals are present simultaneously and noticeably).
- The confidence score is also shown to correlate with the resulting ASR performance, and thus can be used to exclude segments below a threshold from recognition. This also shows that the proposed system essentially tries to segment the data in terms of *recognizable* and *non-recognizable* segments. However, the recognizability and hence the performance of the proposed system for a given sound segment are limited by the (primary) MLP.

In summary, while the proposed technique can also be used for other applications, it provides an efficient and robust pre-processing module for a hybrid HMM/ANN ASR system.

7.2 Speaker Change Detection

A metric based approach to speaker change detection is presented in Chapter 4 where two neighboring windows of relatively small sizes are dynamically moved over the audio signal. The major contribution of this thesis is to propose a novel distance metric which is used to compute the distance between the two windows and accordingly decide if there exists a change point between the two windows. We make the following conclusions from this work:

- Unlike BIC and LLR distance metrics which make use of an adjustable threshold, the proposed distance metric is free of any tuning parameters and still provides similar performance.

- A systematic, although intuitive, analysis of the functioning of the proposed distance metric showed that the metric results in values ranging from positive to negative values for the cases of a speaker change and no speaker change, respectively, making zero a natural threshold for the problem.

- Most of the significant speaker change points are detected within a very small range (order of 400ms). However, the system performance for a true reference change point depends a lot on its *detectability* (distance in time from the next true change point).

7.3 Speaker Clustering

Chapter 5 proposed a speaker clustering algorithm which is efficient and requires minimal human interaction. The algorithm is based on an HMM framework and runs in an agglomerative manner. Starting with a large number of clusters, the algorithm identifies pairs of *closest* clusters and merges them in an iterative manner. The major contribution of this chapter is to define a novel merging criterion to decide if two clusters should be merged or not. Several experiments, with different settings, datasets and feature vectors were carried out to examine the efficiency and robustness of the algorithm. We make the following conclusions from this work:

- The proposed merging criterion is shown to be free of any "tunable" parameters and still outperform the LLR, which requires a threshold value to make a decision about the merging of two clusters.

- The algorithm converges when there are no more pairs of clusters left for merging, eliminating the need for an external method of controlling the clustering process such as heuristics or adjustable threshold values.

- The study of system performance as a function of basic system settings involved in the system illustrated the robustness of the algorithm, as long as the values of these parameters were selected to be in a reasonable range.

- Our experiments with different commonly used feature vectors showed that although all feature vectors perform reasonably well, MFCC coefficients provide the best performance. This is also in accordance with the choice of feature vectors made in most state-of-the-art speaker recognition systems.

- The performance of the algorithm degrades with increasing number of reference speakers in the data. In the case of a limited number of speakers, the number of clusters found by the algorithm often corresponds to the number of reference speakers. When the number of speakers is too large, the algorithm tries to form *pure* clusters for dominant speakers.

- The proposed merging criterion fails to merge clusters having non-speech data and ends up forming a few extra clusters for non-speech data.

- The algorithm showed very competitive performance compared to other state-of-the-art approaches in NIST "who spoke when" speaker diarization evaluations. Moreover, we note that the system often results in high speaker *purities*, a feature that is desirable in applications such as speaker adaptation for ASR.

7.4 Integration

The three tasks investigated in this thesis individually may have to be integrated and/or modified in a practical application. This issue was explored in Chapter 6, where two practical applications are built around these basic tasks. The first application addressed the problem of online audio indexing, where the goal was to first segment the incoming audio stream in terms of homogenous regions, then classify each segment as speech, music or *mixture*, and finally provide consistent labeling (in terms of speakers, speaker clustering) to the speech regions. In this case, the performance of individual modules is clearly dependent on other modules. When compared with the performance of corresponding "independent" module on the same data (broadcast news), we observed the following:

- As expected, some of the insertions made in the segmentation step, especially in the pure music regions, were corrected by the following speech/non-speech discrimination and speaker clustering tasks.

- The performance of the online segmentation system in terms of speech/non-speech classification is inferior compared to that of the offline system, especially for detecting music segments classes correctly. This is because of the inclusion of the *mixture* class, which is considered as speech for the evaluation of the online system.

- The performance of the speaker clustering was very similar to the performance of the offline system presented in Chapter 5. However, we note that the clustering in an online framework as presented in Chapter 6 only deals with clean speech segments as all the non-speech and *mixture* segments are excluded from clustering process. On the other hand, the *offline* clustering algorithm deals with all the data. A similar performance of the two frameworks shows that in *offline* framework, non-speech segments are assigned to clusters other than speech clusters.

The second application in Chapter 6 addressed the problems of clustering the data both in terms of speakers as well as their locations in a meeting room recording. The motivation behind this work was to use the clustering algorithm proposed in Chapter 5 with additional information in terms of precise location based segmentation (Lathoud and McCowan, 2003), available in scenarios like Smart Meeting Rooms. This approach not only results in joint segmentation of the data in terms of "who is speaking and at what place?" but also improves the speaker clustering performance compared to the clustering done using only acoustic cues.

7.5 Future Directions

It was mentioned earlier that, in general, audio segmentation is task of partitioning an audio stream in terms of homogeneous regions, where the rule of homogeneity depends on the application. In this thesis, three significant tasks related to audio segmentation were explored. However, there are various other ways in which an audio stream can be segmented in terms of homogenous regions, *e.g.* languages, environments, music-types, genders, channels, etc. In an extreme case, automatic segmentation in terms of basic sound categories (*e.g.* phonetic, syllabic etc.) may also be considered. In the following, I mention two such ways of segmenting the audio data, which can be approached using basic principles used in this thesis.

Language Identification: We conclude from the work in this thesis and specifically in Chapter 3 that the proposed speech/music segmentation technique essentially tries to segment the data in terms of recognizable and *non-recognizable* segments, where the recognizability is determined (or limited) by the ASR system. If we have hybrid HMM/ANN ASR for several languages and wish to have a language identifier as a preprocessing module, methodology similar to our speech/music segmentation can be employed.

Specifically, the entropy and dynamism features analyzed in this thesis for the purpose of speech/music segmentation can also be investigated for the purpose of language identification. One way to achieve this would be to concatenate the entropy and dynamism feature vectors from individual language ASR systems and make PDFs of these features for each language. This could then be used either in a HMM framework like the one used in this thesis or in hypothesis testing done using LLR.

Automatic Subword Unit Extraction: Subword units can be defined as basic sound units used to represent words in ASR methodology. Almost all state-of-the-art techniques use *phonemes* or units derived from phonemes as subword units. Phonemes are based on "expert" human linguistic knowledge and rely on human perception rather than acoustic observations (data). In an otherwise statistical framework of ASR subword units and a lexicon based on these units are the only components that are not decided by data. I strongly believe that subword units extracted directly from the data would better handle the co-articulation effects within a word and pronunciation variability across different speakers and hence would result in better ASR performance.

Several researchers have addressed this problem in literature as one of segmentation and clustering (Bacchiani and Ostendorf, 1999; Singh *et al.*, 2002; Svendsen *et al.*, 1995). In this formulation of the problem, the acoustic data is first segmented in terms of homogenous "phonetic" regions and then these segments are clustered to provide a consistent subword unit index. Although, the performance achieved in previous work related to subword units is promising (in fact better than phoneme based systems in some cases), no clear conclusions have been made about the potential of *automatic* subword units in ASR. I feel that the reason for this is that most of the previous work was based on heuristics such as pre-determined number of clusters or linear pronunciation length of a word (Bacchiani and Ostendorf, 1999; Bahl *et al.*, 1993). These heuristics forbid a systematic analysis of the system performance as a function of parameters in the system and hence deriving clear conclusions about the potential of subword units in general is difficult.

I tried to use the threshold and heuristic free segmentation and clustering algorithm proposed in this thesis for this purpose and realized that the subword unit clustering problem is different to speaker clustering problems in many aspects. Although I observed similar behavior of the algorithm (in terms of likelihood) in the context of subword unit clustering, it did not translate in ASR performance in terms of WER. However, I believe that a more detailed analysis of the problem and appropriate modifications of the algorithm would result in efficient functioning of the algorithm for this task. This may involve, for example, investigation of optimality criterion other than maximum likelihood, like maximum mutual information (MMI) or WER on a held-out dataset, etc. Another thing I observed and I strongly believe in, is speaker dependent subword units. It is well known that acoustic features like MFCC are highly influenced by the speaker identity, even after sufficient smoothing is done to eliminate this. Thus, it may be a good idea to first cluster the data (automatically) in terms of speakers and then cluster the data within one speaker cluster in terms of subword units. These speaker dependent subword units may better capture the pronunciation pattern and co-articulation effects within a speaker, and hence result in better ASR performance. This can then be extended to the speaker independent case by investigating correlations between subword units across different speakers.

In summary, I feel that automatic subword units is an open, challenging and promising research issue and the techniques proposed in this thesis can be appropriately modified and used to approach this.

Investigation of other threshold free techniques: As mentioned earlier, apart from the three major tasks addressed in this thesis, there are various other ways in which an audio stream can be segmented and classified (*e.g.*environments, music-types, genders, channels, etc.). Investigating the application of threshold free techniques proposed in this thesis for various other audio segmentation tasks would be fruitful.

There are tasks other than audio segmentation such as speaker verification and keyword spotting, which rely a lot on thresholding or other parameter tuning. In my opinion, similar investigation of threshold free techniques for these tasks is also required to make these technologies robust across datasets and acoustic conditions.

Bibliography

Ajmera, J. and Wooters, C. (2003). A robust speaker clustering algorithm. In *Automatic Speech Recognition Understanding Workshop (ASRU)*, pages 411–416.

Ajmera, J., Bourlard, H., Lapidot, I., and McCowan, I. (2002). Unknown-multiple speaker clustering using HMM. In *Int. Conf. on Spoken Language Processing (ICSLP)*, pages 573–576.

Ajmera, J., McCowan, I., and Bourlard, H. (2003). Speech/music segmentation using entropy and dynamism features in a HMM classification framework. *Speech Communication*, **40**, 351–363.

Ajmera, J., Lathoud, G., and McCowan, I. (2004a). Clustering and segmenting speakers and their locations in meetings. In *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*.

Ajmera, J., McCowan, I., and Bourlard, H. (2004b). Robust speaker change detection. *IEEE signal processing letters*. To appear. Available from: ftp://ftp.idiap.ch/pub/reports/2002/rr02-39.ps.gz.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, **19**, 716–723.

Andersen, J. M. (1998). Baseline system for hybrid speech recognition on french (experiments on BREF). IDIAP-COM 07, IDIAP.

ASSAVID Project (2000). Automatic segmentation and semantic annotation of sports videos. http://viplab.dsi.unifi.it/ASSAVID/.

Atal, B. (1976). Automatic recognition of speaker from their voices. *Proc. of the IEEE*, **64**, 460–475.

Atal, B. A. (1974). Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. *Journal of the Acoustical Society of America*, pages 1304–1312.

Bacchiani, M. and Ostendorf, M. (1999). Joint lexicon, acoustic unit inventory and model design. *Speech Communication*, **29**, 99–114.

Bahl, L. R., Brown, P. F., de Souza, P. V., Mercer, R. L., and Picheny, M. A. (1993). A method for the construction of acoustic markov models for words. *IEEE Transactions on Speech and Audio processing*, **1(4)**, 443–452.

Bakis, R. (1997). Transcription of broadcast news shows with the IBM large vocabulary speech recognition system. *Proceedings of the speech recognition workshop*, pages 67–72.

Biernacki, C., Celeux, G., and Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**, 719–725.

Bimbot, F. and Mathan, L. (1993). Text-free speaker recognition using an arithmetic harmonic sphericity measure. In *European Conf. on Speech Communications and Technology (EUROSPEECH)*, volume 1, pages 169–172.

Bishop, C. M. (1996). *Neural networks for pattern recognition*. Oxford university press, UK.

Blum, T. L., Keislar, D. F., Wheaton, J. A., and Wold, E. H. (1999). Method and article of manufacture for content-based analysis, storage, retrieval, and segmentation of audio information. *U. S. Patent 5*.

Bonastre, J. F., Delacourt, P., Fredouille, C., Merlin, T., and Wellekens, C. (2000). A speaker tracking system based on speaker turn detection for NIST evaluations. In *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1177–1180.

Bourlard, H. and Morgan, N. (1994). *Connectionist Speech Recognition*. Kluwer Academic Press.

Chen, S. S. and Gopalakrishnan, P. S. (1998). Speaker, environment and channel change detection and clustering via the Bayesian information criterion. *IBM Technical Journal*. Available from: http://www.nist.gov/speech/publications/darpa98/html/bn20/bn20.htm.

Christmas, W., Jaser, E., Messer, K., and Kittler, J. (2003). A multimedia system architecture for automatic annotation of sports videos. In J. C. et al., editor, *Computer vision systems: Third International Conference on Computer Vision Systems*, volume 2626 of *LNCS*, pages 513 – 522. Springer.

CIMWOS Project (2000). Combined image and word spotting. http://www.xanthi.ilsp.gr/cimwos/.

Cover, T. M. and Thomas, J. A. (1991). *Elements of information theory*. New York: John Wiley and Sons.

Cutler, R., Rui, Y., Gupta, A., Cadiz, J., Tashev, I., He, L., Colburn, A., Zhang, Z., Liu, Z., and Silverbeg, S. (2002). Distributed meetings : A meeting capture and broadcasting system. In *Proc. of ACM Multimedia*.

Davis, S. B. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken utterances. *IEEE Transactions on Speech and Audio Processing*, **28**, 357–366.

Delacourt, P. and Wellekens, C. J. (2000). DISTBIC: A speaker based segmentation for audio data indexing. *Speech Communication*, **32**, 111–126.

Delacourt, P., Kryze, D., and Wellekens, C. J. (1999). Detection of speaker changes in an audio document. In *European Conf. on Speech Communications and Technology (EUROSPEECH)*, pages 1195–1198.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, **39**(1), 1–38.

DiBiase, J., Silverman, H., and Brandstein, M. (2001). Robust localization in reverberant rooms. In M. Brandstein and D. Ward, editors, *Microphone Arrays*, chapter 8, pages 157–180. Springer.

Edward, A. W. F. (1972). *Likelihood*. Cambridge University Press.

Foote, J. (1999). An overview of audio information retrieval. *Multimedia Systems*, **7**(1), 2–10. Available from: http://citeseer.ist.psu.edu/foote98overview.html.

Fukunaga, K. (1990). *Statistical Pattern Recognition*. Academic Press.

Gauvain, J. L. and Barras, C. (2003). EARS RT-03 diarization, limsi-cnrs. *RT-03S Workshop, Boston, MA, USA*. Available from: http://www.nist.gov/speech/tests/rt/rt2003/spring/presentations/limsi_rt0503spkrvg.pdf.

Gish, H. and Schmidt, H. (1994). Text independent speaker identification. *IEEE Signal Processing Magazine*, pages 18–21.

Gish, H., Siu, M. H., and Rohlicek, R. (1991). Segregation of speakers for speech recognition and speaker identification. In *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 873–876.

Hain, T., Johnson, S. E., Turek, A., Woodland, P. C., and Young, S. J. (1998). Segment generation and clustering in the HTK broadcast news transcription system. *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, pages 133–137.

Hermansky, H. (1990). Perceptual linear predictive (plp) analysis of speech. *Journal of the Acoustical Society of America,*, **87**, 1738–1752.

Hermansky, H. and Morgan, N. (1994). RASTA processing of speech. *IEEE Trans. on Speech and Audio Processing*, **2**(4), 578–89.

Jain, U., Siegler, M. . A., Doh, S. J., Huerta, J., Moreno, P. J., Raj, B., and Stern, R. M. (1996). Recognition of continuous broadcast news with multiple unknown speakers and environments. *DARPA Speech Recognition Workshop*.

Jin, H., Kubala, F., and Schwartz, R. (1997). Automatic speaker clustering. *Proceedings of the DARPA speech recognition workshop*.

Johnson, S. (1999). Who spoke when? - automatic segmentation and clustering for determining speaker turns. In *European Conf. on Speech Communications and Technology (EUROSPEECH)*, volume 5, pages pp. 2211–2214, Budapest,Hungary. Available from: http://citeseer.nj.nec.com/johnson99who.html.

Johnson, S. and Woodland, P. (1998). Speaker clustering using direct maximization of the MLLR-adapted likelihood. In *Int. Conf. on Spoken Language Processing (ICSLP)*, volume 5, pages 1775–1778, Sydney, Australia. Available from: http://citeseer.nj.nec.com/21363.html.

K. El-Maleh, M. Klein, G. P. and Kabal, P. (2000). Speech/ music discrimination for multimedia application. In *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2445–2448.

Kemp, T., Schmidt, M., Westphal, M., and Waibel, A. (2000). Strategies for automatic segmentation of audio data. In *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, volume 3, pages 1423–1426.

Kubala, F. (1997). The 1996 BBN byblos hub-4 transcription system. *Proceedings of the speech recognition workshop*, pages 90–93.

Lapidot, I. (2003). SOM as likelihood estimator for speaker clustering. *European Conf. on Speech Communications and Technology (EUROSPEECH)*.

Lathoud, G. and McCowan, I. (2003). Location based speaker segmentation. In *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*.

Liu, D. and Kubala, F. (1999). Fast speaker change detection for broadcast news transcription and indexing. In *European Conf. on Speech Communications and Technology (EUROSPEECH)*, pages 1031–1034.

Logan, B. (2000). Mel frequency cepstral coefficients for music modeling. *International Symposium on Music Information Retrieval*, **28**. Available from: http://ismir2000.ismir.net/papers/logan_paper.pdf.

Logan, B. T. and Chu, S. (2000). Music summarization using key phrases. In *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*.

Lopez, J. F. and Ellis, D. P. W. (2000). Using acoustic condition clustering to improve acoustic change detection on broadcast news. In *Int. Conf. on Spoken Language Processing (ICSLP)*.

M4 Project (2002). Multi modal meeting manager. http://www.m4project.org/.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability,*, **1**, 281–297.

Makhoul, J. (1975). Linear prediction: A tutorial review. *Proc. of the IEEE*, **63**, 561–580.

Martyn, A., Przybocki, M., Doddington, G., and Reynolds, D. (2000). The nist speaker recognition evaluation - overview, methodology, systems,results, perspectives (1998). *Speech Communications*, pages 225–254. Available from: http://www.nist.gov/speech/publications/papersrc/spkoverview.pdf.

Matsoukas, S., Iyer, R., Kimball, O., Ma, J., and Colthurst, T. (2003). BBN CTS english system. *RT-03S Workshop, Boston, MA, USA*. Available from: http://www.nist.gov/speech/tests/rt/rt2003/spring/presentations/cts-combined-sm-ok-v14.pdf.

McCowan, I., Bengio, S., Gatica-Perez, D., Lathoud, G., Monay, F., Moore, D., Wellner, P., and Bourlard, H. (2003a). Modeling human interactions in meetings. In *Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

McCowan, I., Ajmera, J., and Moore, D. (2003b). An online system for automatic annotation of audio documents. IDIAP-RR 39, IDIAP.

Misra, H., Ikbal, S., and Yegnanarayana, B. (2003). Speaker-specific mapping for text-independent speaker recognition. *Speech Communications*, pages 301–310.

Moore, D. (2002). The IDIAP Smart Meeting Room. IDIAP-COM 07, IDIAP.

Moraru, D., Meignier, S., Fredouille, C., and Bonastre, J. F. (2003). ELISA, CLIPS and LIA NIST 2003 segmentation. *RT-03S Workshop, Boston, MA, USA*. Available from: http://www.nist.gov/speech/tests/rt/rt2003/spring/presentations/Nist2003-LIA-CLIPS-segv2.pdf.

Moreno, P. and Rifkin, R. (2000). Using the fisher kernel method for web audio classification. In *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1921–1924.

Mori, K. and Nakagawa, S. (2002). Speaker change detection and speaker clustering using VQ distortion for broadcast news speech recognition. In *Int. Conf. on Pattern Recognition (ICPR)*.

Nguyen, P. (2003). PSTL's speaker diarization. *RT-03S Workshop, Boston, MA, USA*. Available from: http://www.nist.gov/speech/tests/rt/rt2003/spring/presentations/isometspkr.pdf.

Papoulis, A. (1991). *Probability, Random Variable and Stochastic Processes*. McGraw Hill, NY.

Rabiner, L. and Juang, B.-H. (1993). *Fundamentals of Speech Recognition*. Prentice Hall Signal Processing Series.

Ramabhadran, B., Huang, J., Chaudhari, U., Iyengar, G., and Nock, H. J. (2003). Impact of audio segmentation and segment clustering on automated transcription accuracy of large spoken archives. In *European Conf. on Speech Communications and Technology (EUROSPEECH)*, pages 2589–2592.

Reynold, D., Torres, P., and Roy, R. (2003). EARS RT-03 diarization, MIT lincoln laboratory. *RT-03S Workshop, Boston, MA, USA*. Available from: http://www.nist.gov/speech/tests/rt/rt2003/spring/presentations/index.htm/EARSRT03SDiarization.pdf.

Reynolds, D., Quatieri, T. F., and Dunn, R. B. (2000). Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, **10**(1–3).

Rissanen, J. (1989). Stochastic complexity in statistical inquiry. *Singapore: World Scientific*.

Saunders, J. (1996). Real-time discrimination of broadcast speech/ music. In *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 993–996.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, **6**, 461–464.

Seck, M., Bimbot, F., Zugah, D., and Delyon, B. (1999). Two-class signal segmentation for speech/music detection in audio tracks. In *European Conf. on Speech Communications and Technology (EUROSPEECH)*.

Sheirer, E. and Slaney, M. (1997). Construction and evaluation of a robust multifeature speech/ music discriminator. In *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1331–1334.

Siegler, M. . A., Jain, U., Raj, B., and Stern, R. M. (1997). Automatic segmentation, classification and clustering of broadcast news. *DARPA Speech Recognition Workshop, Chantilly*, pages 97–99.

Singh, R., Raj, B., and Stern, R. M. (2002). Automatic generation of subword units for speech recognition systems. *IEEE Transactions on Speech and Audio Processing*, **10(2)**, 89–99.

Solomonoff, A., Mielke, A., Schmidt, M., and Gish, H. (1998). Clustering speakers by their voices. *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 757–760.

Spina, M. S. and Zue, V. W. (1996). Automatic transcription of general audio data. In *Int. Conf. on Spoken Language Processing (ICSLP)*, Philadelphia, USA.

Stolcke, A., Franco, H., Gadde, R., and Graciarena, M. (2003). Speech-to-text research at SRI-ICSI-UW. *RT-03S Workshop, Boston, MA, USA*. Available from: http://www.nist.gov/speech/tests/rt/rt2003/spring/presentations/index.htm/sri+-rt03-stt.pdf.

Svendsen, T., Soong, F. K., and Purnhagen, H. (1995). Optimizing baseforms for HMM based speech recognition. In *European Conf. on Speech Communications and Technology (EUROSPEECH)*, pages 783–786.

THISL Project (1997). Thematic indexing of spoken language. http://www.dcs.shef.ac.uk/research/groups/spandh/projects/thisl/.

Tranter, S., Yu, K., and the HTK STT team (2003). Diarization for RT-03s at cambridge university. *RT-03S Workshop, Boston, MA, USA*. Available from: http://www.nist.gov/speech/tests/rt/rt2003/spring/presentations/tranter_rt03sdiary.2up.pdf.

Tritschler, A. (1998). A segmentation enabled speech recognition application using BIC. *M. S. Thesis, Institute Eurecom (France)*.

Tritschler, A. and Gopinath, R. (1999). Improved speaker segmentation and segments clustering using the Bayesian information criterion. In *European Conf. on Speech Communications and Technology (EUROSPEECH)*, pages 679–682.

Vandecatseye, A. and Martens, J. P. (2003). A fast, accurate and stream-based speaker segmentation and clustering algorithm. *European Conf. on Speech Communications and Technology (EUROSPEECH)*.

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Trans. Information Theory*, **IT-13**, 260–269.

Vuuren, S. (1996). Comparison of text-independent speaker recognition methods on telephone speech with acoustic mismatch. In *Int. Conf. on Spoken Language Processing (ICSLP)*, pages 1788–1792.

Wang, Y., Liu, Z., and Huang, J.-C. (2000). Multimedia content analysis. *IEEE Signal Processing Magazine*, pages 12–36.

Williams, G. and Ellis, D. (1999). Speech/ music discrimination based on posterior probabilities. In *European Conf. on Speech Communications. and Technology (EUROSPEECH)*, pages 687–690.

Woodland, P., Gales, M., Pye, D., and Young, S. (1997). The development of the 1996 HTK broadcast news transcription system. *Proceedings of the speech recognition workshop*, pages 73–78.

Woodland, P., Evermann, G., Gales, M., Hain, T., Chan, R., and Jia, B. (2003). CU-HTK STT system for RT-03. *RT-03S Workshop, Boston, MA, USA*. Available from: http://www.nist.gov/speech/tests/rt/rt2003/spring/presentations/stt-overall.pdf.

Zhou, B. and Hansen, J. H. L. (2000). Unsupervised audio stream segmentation and clustering via the bayesian information criterion. In *Int. Conf. on Spoken Language Processing (ICSLP)*, pages 714–717.

Curriculum Vitae

Jitendra Ajmera

Permanent address:	1-I-13, Mahaveer Nagar 3^{rd}	Phone:	+91 7451 220536
	Kota, Rajasthan	email:	Jitendra@idiap.ch
	324005-India	Citizenship:	Indian

Work Experience

2001–	Dalle Molle Institute for Perceptual Artificial Intelligence (IDIAP), Switzerland Speech Processing Group Research Assistant
1999-2000	Patni Computer Systems, Mumbai, India Research & Development Division Software Engineer

Others

Jan.-June 2003	International Computer Science Institute (ICSI), Berkeley, CA, USA Visiting Researcher

Education

2000–	Docteur ès Sciences (anticipated June 2004) The Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland Dissertation: *Robust Audio Segmentation*.
Jan.-June 2001	Postgraduate Course in Speech and Language Engineering The Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland
1995–1999	Bachelor of Technology in Electrical Engineering Indian Institute of Technology, Bombay, India

Computer Experience

Operating Systems: Solaris, UNIX/Linux, Windows

Computer Applications: HTK 2.1, MATLAB, TCL-TK

Languages: C++, C, Visual C++.

Published/Accepted Papers

Ajmera, J., McCowan, I., and Bourlard, H. (2003). Speech/music segmentation using entropy and dynamism features in a HMM classification framework. *Speech Communication*, **40**, 351–363.

Ajmera, J., McCowan, I., and Bourlard, H. (2004). Robust speaker change detection. *IEEE signal processing letters*. To appear. Available from: ftp://ftp.idiap.ch/pub/reports/2002/rr02-39.ps.gz.

Ajmera, J. and Wooters, C. (2003). A robust speaker clustering algorithm. In *Automatic Speech Recognition Understanding Workshop (ASRU)*, pages 411–416.

Ajmera, J., Lathoud, G., and McCowan, I. (2004). Clustering and segmenting speakers and their locations in meetings. In *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*

Ajmera, J., McCowan, I., and Bourlard, H. (2002). Robust HMM based speech/music segmentation. In *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1746–1749.

Ajmera, J., Bourlard, H., Lapidot, I., and McCowan, I. (2002). Unknown-multiple speaker clustering using HMM. In *Int. Conf. on Spoken Language Processing (ICSLP)*, pages 573–576.

Unpublished Technical Reports

McCowan, I., Ajmera, J., and Moore, D. (2003). An online system for automatic annotation of audio documents. IDIAP-RR 03-39, IDIAP. Available at ftp://ftp.idiap.ch/pub/reports/2003/rr03-39.ps.gz.

Ajmera, J., Bourlard, H., Lapidot, I. (2002). Improved Unknown-multiple speaker clustering using HMM. Available at ftp://ftp.idiap.ch/pub/reports/2002/rr-02-23.ps.gz.