SPEAKER VERIFICATION FOR MICROPHONE SUITABLE DATA AND AUDIO DIARIZATION FOR TURKISH BROADCAST NEWS

by

Oğuz Yılmaz

B.S., Electrical and Electronics Engineering, Boğaziçi University, 2009

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronics Engineering
Boğaziçi University

2011

SPEAKER VERIFICATION FOR MICROPHONE SUITABLE DATA AND AUDIO DIARIZATION FOR TURKISH BROADCAST NEWS

APPROVED BY:

Assoc. Prof. Murat Saraçlar .....
(Thesis Supervisor)

Prof. Levent Arslan .....

Asst. Prof. Cenk Demiroğlu .....

DATE OF APPROVAL: 14.07.2011

ACKNOWLEDGEMENTS

This work would not be possible without help support and assistance from a number of people. I would like to thank the following people. My supervisor, Murat Saraçlar, for always having moment to spare and his encouragement, guidance and support from the initial to the final level of this study. My professors, A.Taylan Cemgil, Bülent Sankur and Kerem Harmancı for everything I learned in their excellent lectures. I will always miss the discussions on various topics with Kerem Harmancı. He was great teacher. He will be always in my thoughts and forever in my heart. Erinç Dikici for having such a helpful colleague and wonderful friend. I could not overcome many responsibilities during last two years without his support. I also thank him for preparation of the GMM/UBM baseline system for NIST SRE 2010. All BUSIM members for the warmful environment in this lab. I am pleased to be a member in this sincere family. My friend, Rüştü Dücan for his continuous support during preperation of my thesis. Speech Processing Group Members at Brno University of Technology (BUT) for letting me to work on state-of-the-art problems during my Erasmus term studies at BUT. Especially, I am grateful for having Ondřej Glembek as my advisor at BUT. His consistent help during this period enabled me to have a deeper understanding of the field. Also, I would like to thank Lukáš Burget for his insightful comments and criticisms for my studies.

I would like to thank TUBITAK for their financial support. This work is financially supported by the COST 2101 project "Biometrics for Identity Documents for Smart Cards", TUBITAK project 107E001 and TUBITAK project 105E102.

Lastly, I would like to express my deepest gratitude to my dear family for their endless love and support.

ABSTRACT

SPEAKER VERIFICATION FOR MICROPHONE SUITABLE DATA AND AUDIO DIARIZATION FOR TURKISH BROADCAST NEWS

In this thesis, speaker verification and audio diarization tasks are studied. The aim of speaker verification is to determine whether two utterances are spoken by same speaker. Investigators from many research group participate in the annual Speaker Recognition Evaluations (SRE) which is organized by the National Institute of Standards and Technology (NIST) in order to analyze the performance of various methods. In 2010, three groups from Turkey including Boğaziçi University participated in the evaluation. Two baseline systems were developed for this evaluation and acceptable system performance was obtained for the first time submission. A problem with SRE 2010 is that development data for microphone case is sparse. Use of sufficient amount of telephone data in conjunction with limited microphone data is investigated to improve system performance of microphone conditions. The diarization is task of explanation of all sources in an audio. Turkish Broadcast News data is utilized in this task. Baseline and factor analysis based systems are developed and a comparative study between these two systems is reported. It has been shown that performance of speech recognition systems can be improved by adaptation of speakers whose data can be obtained via automated audio diarization. A similar study is performed using Turkish Broadcast News data. Lastly, a novel algorithm is proposed for segmentation of simultaneous speech segments. It is shown in the experiments that the proposed approach improved the overall system performance.

ÖZET

MİKROFON VERİSİNE UYGUN KONUŞMACI DOĞRULAMA VE TÜRKÇE HABER PROGRAMLARI İÇİN İŞİTSEL BÖLÜTLEME

Bu tezde konuşmacı doğrulama ve işitsel bölütleme üzerine çalışıldı. Konuşmacı doğrulamada amaç verilen iki söyleyişin aynı konuşmacı tarafından söylenip söylenmediğinin belirlenmesidir. Bu problem için bir çok yöntem önerilmiştir. Bu sistemlerin performasını inceleyebilmek ve son teknoloji yöntemlerini belirleyebilmek amacıyla bir çok araştırma grubundan araştırmacılar, Ulusal Standartlar ve Teknoloji Enstitüsü (NIST) tarafından düzenlenen her yılki Konuşmacı Tanıma Değerlendirmelerine (SRE) katılmaktadır. Türkiye'den 2010 yılında, Boğaziçi Üniversitesi ile birlikte toplam üç grup bu değerlendirmeye katılmıştır. Değerlendirmeye katılmak için iki referans sistemi geliştirilmiş ve ilk katılım için kabul edilebilir sistem sonuçları elde edilmiştir. 2010 yılı değerlendirmelerindeki bir problem mikrofon verisi için geliştirme verisinin kısıtlı olmasıdır. 2010 değerlendirmesinde mikrofon test durumlarındaki sistem performasını artırılması amacıyla yeterli miktarda telefon verisi ile birlikte kısıtlı mikrofon verisinin kullanımı üzerine çalışıldı. Çalışmanın ikinci aşamasında işitsel bölütleme üzerine çalışıldı. Bölütleme verilen bir ses katarındaki bütün kaynakların açıklanmasıdır. Grubumuzca derlenen Türkçe Haber Programları verileri bu amaç için kullanıldı. Bir referans sistemi ile son teknoloji faktör analizi temelli bir işitsel bölütleyici geliştirildi ve bu iki sistemin karşılaştırmalı analizi raporlandı. Konuşmacı verisinin otomatik işitsel bölütleyiciyle elde edildiği konuşmacı uyarlamasının, konuşma işleme sistemi performansının iyileştirdiği bilinmektedir. Benzer bir çalışma Türkçe Haber Bültenleri kullanılarak uygulandı. Son olarak, çakışan konuşmaların bölütlenmesi üzerinde çalışıldı. Bu amaç için yeni bir algoritma önerildi. Önerilen yakalaşımın, sistem performasını iyileştirdiği deneylerde bir miktar çakışan konuşma içeren veriler kullanılarak gösterildi.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . iv
ÖZET . . . v
LIST OF FIGURES . . . ix
LIST OF TABLES . . . xii
LIST OF SYMBOLS . . . xiv
LIST OF ACRONYMS/ABBREVIATIONS . . . xvi
1. INTRODUCTION . . . 1
1.1. Speaker Recognition . . . 1
1.2. Audio Diarization . . . 2
1.3. Main Contributions . . . 3
1.4. Thesis Organization . . . 4
2. BACKGROUND . . . 5
2.1. Front End Processing . . . 5
2.1.1. MFCC Parametrization . . . 6
2.1.2. Channel Compensation in Front End Vectors . . . 8
2.2. Speaker Modeling and Classification . . . 9
2.2.1. Gaussian Mixture Model . . . 9
2.2.1.1. Universal Background Model . . . 12
2.2.1.2. Maximum-a-Posteriori Adaptation . . . 13
2.2.1.3. Maximum Likelihood Linear Regression . . . 14
2.2.2. Support Vector Machines . . . 15
2.2.2.1. GMM Supervector Linear Kernel . . . 17
2.2.3. Joint Factor Analysis . . . 18
2.2.4. I-Vector Analysis . . . 22
2.2.4.1. Scoring Methods for I-Vectors . . . 24
2.2.4.2. Probabilistic Linear Discriminant Analysis . . . 26
3. SPEAKER VERIFICATION . . . 29
3.1. Overview of a Speaker Verification System . . . 29

3.2. Speaker Recognition Methodologies 29
3.2.1. GMM/UBM based Speaker Verification System 30
3.2.2. GMM/SVM based Speaker Verification System 31
3.2.3. Factor Analysis based Speaker Verification System 32
3.3. Score Normalization 34
3.3.1. Zero Normalization 34
3.3.2. Test Normalization 35
3.3.3. Symmetric Normalization 35
3.3.4. Other Normalizations 36
3.4. Evaluation Metric 36
3.5. Speaker Verification Experiments 39
3.5.1. Baseline Systems for NIST 2010 SRE 39
3.5.1.1. GMM/UBM Baseline System 39
3.5.1.2. GMM/SVM Baseline System 40
3.5.1.3. GMM/UBM and SVM Fusion 41
3.5.1.4. Results 41
3.5.2. I-vector Analysis with PLDA Scoring for Microphone Data . . . 43
4. AUDIO DIARIZATION 49
4.1. Baseline Audio Diarization Setup 49
4.1.1. Speech Detection 49
4.1.2. Speaker Turn Point Detection 50
4.1.3. Clustering 52
4.1.4. Re-segmentation 53
4.2. Overlapping Speaker Segmentation 53
4.2.1. Maximum Likelihood Speech Source Separation 54
4.2.2. Overlap Detection 55
4.3. Factor Analysis Based Audio Diarization 56
4.4. Speaker Adaptation for LVCSR via MLLR 58
4.5. Error Metric 58
4.6. Experiments 59
4.6.1. Speaker Segmentation Setups 59

4.6.2. Speaker Adaptation Experiments 60
4.6.2.1. Experimental Setup 60
4.6.2.2. Results 61
4.6.3. Experiments with Overlapping Speaker Segmentation 64
4.6.3.1. Results 64
4.6.4. Experiments with Factor Analysis Based Systems 66
4.6.4.1. System Modeling 66
4.6.4.2. Systems 68
4.6.4.3. Results for More Than Two Speakers Segmentation . . 69
4.6.4.4. Results for Two Speakers Segmentation 70
5. CONCLUSION 72
APPENDIX A: DETAILED RESULTS OF MICROPHONE CONDITIONS . 75
APPENDIX B: RESULTS OF BOUN SUBMISSION FOR NIST SRE 2010 . . 90
REFERENCES 94

LIST OF FIGURES

Figure 2.1. The main blocks of speaker verification. 5

Figure 2.2. The main blocks of audio diarization. 6

Figure 2.3. MFCC computation scheme. 7

Figure 2.4. A typical example of a SVM. 16

Figure 2.5. Graphical model of Gaussian PLDA. 27

Figure 3.1. Block diagram of speaker verification system. 30

Figure 3.2. Block diagram of GMM/UBM system. 31

Figure 3.3. Block diagram of GMM/SVM system. 32

Figure 3.4. Block diagram of microphone suitable I-vector estimation. 33

Figure 3.5. An example of a DET curve. 37

Figure 3.6. DET curve for core condition 1. 42

Figure 3.7. DET curve for core condition 5. 42

Figure 3.8. Constellation plot for all results in condition 1. 45

Figure 3.9. Constellation plot for all results in condition 2. 46

Figure 3.10. Constellation plot for all results in condition 3. 46

Figure 3.11. Constellation plot for all results in condition 4. 47

Figure 3.12. Constellation plot for all results in condition 5. 47

Figure 4.1. Illustration of HMM based speech detection. 50

Figure 4.2. Illustration of BIC turn point detection. 51

Figure 4.3. Illustration of HAC. 52

Figure 4.4. Illustration of Viterbi re-segmentation. 53

Figure 4.5. Speaker adaptation results for setup 1. 62

Figure 4.6. Speaker adaptation results for setup 2. 62

Figure 4.7. Speaker adaptation results for setup 3. 63

Figure 4.8. Fully automatic segmentation results for all experiments. 63

Figure B.1. Results for core-core condition 2. 90

Figure B.2. Results for core-core condition 3. 91

Figure B.3. Results for core-core condition 4. 91

Figure B.4. Results for core-core condition 6. 92

Figure B.5. Results for core-core condition 7. 92

Figure B.6. Results for core-core condition 8. 93

Figure B.7. Results for core-core condition 9. 93

LIST OF TABLES

Table 3.1. Cost and prior parameters used in NIST SRE 2010. 38

Table 3.2. Training schemes for I-vector extractor. 43

Table 3.3. I-vector scoring methods with various training data conditions. . . 44

Table 3.4. Types of data in all conditions. 45

Table 4.1. Summary of the test instance by time. 65

Table 4.2. Result of speech / non-speech detection. 65

Table 4.3. Results of the three experimental setups. 65

Table 4.4. Detection error rates of HAC systems (%). 70

Table 4.5. Detection error rates of soft speaker segmentation systems. 71

Table 4.6. Detection error rates of soft speaker segmentation systems after Viterbi. 71

Table A.1. Results of Tel400 I-vector scoring methods for condition 1. 75

Table A.2. Results of Tel400Int200 I-vector scoring methods for condition 1. . 76

Table A.3. Results of Tel400Tel200 I-vector scoring methods for condition 1. . 77

Table A.4. Results of Tel400 I-vector scoring methods for condition 2. 78

Table A.5. Results of Tel400Int200 I-vector scoring methods for condition 2. . 79

Table A.6. Results of Tel400Tel200 I-vector scoring methods for condition 2. . 80

Table A.7. Results of Tel400 I-vector scoring methods for condition 3. 81

Table A.8. Results of Tel400Int200 I-vector scoring methods for condition 3. . 82

Table A.9. Results of Tel400Tel200 I-vector scoring methods for condition 3. . 83

Table A.10. Results of Tel400 I-vector scoring methods for condition 4. 84

Table A.11. Results of Tel400Int200 I-vector scoring methods for condition 4. . 85

Table A.12. Results of Tel400Tel200 I-vector scoring methods for condition 4. . 86

Table A.13. Results of Tel400 I-vector scoring methods for condition 5. 87

Table A.14. Results of Tel400Int200 I-vector scoring methods for condition 5. . 88

Table A.15. Results of Tel400Tel200 I-vector scoring methods for condition 5. . 89

LIST OF SYMBOLS

$\mathbf{0}$	Zero vector
$\mathbf{A}$	LDA projection matrix
$\mathbf{B}$	Cholesky decomposition of within class covariance matrix
D	Dimension of acoustic vector
$\mathbf{D}$	JFA residual speaker variability matrix
$D(p(.)\|\|p(.))$	KL distance
$f(.)$	SVM function
f_{lin}	Linear frequency
f_{mel}	Mel frequency
$\overline{\mathbf{F}}$	Supervector obtained by concatenating $\overline{\mathbf{F}}_c$ $(c = 1, \ldots, C)$
$\mathbf{F}_c$	First order statistic for mixture c
$\overline{\mathbf{F}}_c$	Mean centered first order statistic for mixture c
$E[.]$	Excepted value of a random variable
$\mathbf{I}$	Identity matrix
K	Number of samples
$k(.,.)$	Cosine kernel
$K(.,.)$	SVM kernel
$\mathbf{N}$	Diagonal matrix whose diagonals $N_c\mathbf{I}$ $(c = 1, \ldots, C)$
$\mathcal{N}$	Normal distribution
N_c	Zero order statistic for mixture c
$\mathbf{P}$	NAP projection matrix
$p(.)$	Probability of a random variable
$\mathbf{R}$	NAP channel space
$\overline{\mathbf{S}}$	Diagonal matrix whose diagonals $\overline{\mathbf{S}}_c$ $(c = 1, \ldots, C)$
$\mathbf{S}_c$	Second order statistic for mixture c
$\overline{\mathbf{S}}_c$	Mean centered second order statistic for mixture c
$\mathbf{T}$	Total variability matrix
$\mathbf{T}_c$	c^{th} component of total variability matrix

$\mathbf{U}$	JFA channel variability matrix
$\mathbf{U_1}$	PLDA eigenvoice matrix
$\mathbf{U_2}$	PLDA eigenchannel matrix
$\mathbf{V}$	JFA speaker variability matrix
$\mathbf{W}$	Within class covariance matrix
$\mathbf{w}$	I-vector
$\mathbf{Y}$	Adaptation matrix
α_c	GMM adaptation coefficient for mixture c
α_i	SVM Lagrange multipliers
γ	Scaling factor
θ	Mean and covariance parameters of GMM
μ	Mean vector
$\mathbf{\Sigma}$	Covariance matrix
ω	GMM weight parameter

LIST OF ACRONYMS/ABBREVIATIONS

ASR	Automatic Speech Recognition
BIC	Bayesian Information Criteria
BUT	Brno University of Technology
CLR	Cross Likelihood Ratio
CMS	Cepstral Mean Subtraction
DCF	Decision Cost Function
DCT	Discrete Cosine Transform
DET	Detection Error Trade-off
EER	Equal Error Rate
EM	Expectation Maximization
GMM	Gaussian Mixture Model
HAC	Hierarchical Agglomerative Clustering
HLDA	Heteroscedastic Linear Discriminant Analysis
HMM	Hidden Markov Models
JFA	Joint Factor Analysis
KL	Kullback Leibler
LDA	Linear Discriminant Analysis
LPC	Linear Predictive Coding
LVCSR	Large Vocabulary Continuous Speech Recognition
MAP	Maximum a Posteriori
MFCC	Mel Frequency Cepstral Coefficients
MLLR	Maximum Likelihood Linear Regression
NAP	Nuisance Attribute Projection
NIST	National Institute of Standards and Technology
PLDA	Probabilistic Linear Discriminant Analysis
PLP	Perceptual Linear Predictive Coding
ROC	Receiver Operating Curve
RT	Rich Transcription

RTF	Real Time Factor
S-Norm	Symmetric Normalization
SRE	Speaker Recognition Evaluations
STFT	Short Time Fourier Transform
SVM	Support Vector Machines
T-Norm	Test Normalization
UBM	Universal Background Model
WCCN	Within Class Covariance Normalization
WER	Word Error Rate
Z-Norm	Zero Normalization

1. INTRODUCTION

In this thesis, two main topics are studied, speaker recognition and audio diarization. Although practical use and applications of these topics can be quite different, the underlying systems and statistical approaches to these tasks are quite similar, thus enabling to study both of them in a single thesis. In the following sections, an introduction to these topics, contribution of this thesis to the field and organization of the thesis are presented.

1.1. Speaker Recognition

Biometric techniques are concerned with identification of a person or verification of a person identity based on physiological or any other human related characteristics. Biometric verification entails a comparison between one or more enrolled biometric samples and another sample that is given during recognition time. In speaker recognition, acoustic features are utilized as characteristics of an invariant human trait. The unique information that is embedded in acoustic features of each person is due to many effects like anatomical differences in shape of month or throat, voice pitch or speaking style although some of them may vary in time, i.e aging effects or different emotional conditions.

Speaker recognition is referred for both identification and verification tasks. Identification is a closed set problem and the aim is to determine the identity of a speaker in a given closed set. On the other hand, speaker identity is made known to the system in the verification task. The only requirement in both of the tasks is use of a microphone, thus it enables a cost effective solution for long distance recognition applications. Speech input can be enrolled to speaker recognition systems in two different ways, text-dependent or text-independent. In text-dependent case, input speech is selected from foreknown password or any other phrases. However, text independent speaker verification, where speaker enrolls input speech under no constraint, gained more interest in the literature and and it is studied annually in NIST SRE. For these

reasons, only text independent speaker verification is studied in this thesis.

The classical approach in speaker verification uses a probabilistic model, Gaussian Mixture Model (GMM), to represent the distribution of speech that is related to each speaker. The classical solution in classification is based on log-likelihood ratio between speaker and non-speaker model given the speech input. Variants of systems offered later in literature made use of the GMM approach. One of the successful approaches is discriminative modeling of speakers using Support Vector Machines (SVM). Factor Analysis approaches currently have state-of-the-art performance on speaker verification task via modeling latent channel and speaker variables.

1.2. Audio Diarization

Indexating, searching and accessing information in vast amounts of audio including broadcasts, meetings, telephone conversations and other audio sources require automated efficient solutions. Automated transcription of data can provide a base solution to the problem but continuous sequence of words that are contained in the audio can be hard to read. More information like sentence boundaries, speaker turn points and detection of speaker identities can be extracted to make the document more readable. These problems are studied within the context of audio diarization. More formally, audio diarization is the task of categorizing the audio sources in a spoken document. Apart from enrichment of transcription of text, audio diarization potentially helps with other tasks such as continuous speech recognition and machine translation via providing pre-information about content of sources. The simplest diarization is detection of speech and non-speech when all sources is defined as music and noise or silence. More detailed diarization requires categorization of each different speakers with their associated speech segment boundaries as well as categorization of any non speech content.

Audio diarization is used in three main areas, telephone conversations, recorded meetings and broadcast news. The quality and content of the data varies, thus presenting unique diarization challenges for each case. The quality of the data differs due

to transmission bandwidth, microphone quality, noise level and any other differences.

1.3. Main Contributions

This thesis has contributed to both speaker verification and diarization in some aspects.

Study on speaker verification presented in this thesis consists of two mainframes. In the first one, two baseline speaker verification systems, which are based on classical and discriminative approaches, are developed in the first time participation to NIST SRE 2010 from Boğaziçi University as well as first time in Turkey with two other groups, Özyeğin University and Tübitak UEKAE/Sabanci University. Satisfactory performance for a first time submission is achieved with the baseline systems.

In the second part, a more specialized problem than just establishing baseline systems is studied. Lately, use of microphone data in NIST SREs is included. However, microphone data for training is not as abundant as telephone data and this posed a new challenge, training with sparse microphone data. Currently available microphone data is not sufficient in training the state-of-the-art factor analysis based speaker verification. Possible use of different data combinations, telephone, microphone and interview in the training phase to avoid sparse data problem is studied in this thesis. Improvement over the factor analysis based setup is achieved by using the proposed configurations of the training data.

Three different points are studied for the audio diarization task. First, a baseline audio diarization system, that is utilized in all experiments, is developed for Turkish Broadcast News data. Speaker segmentation information of audio, that is obtained using automated audio diarization, helps in reducing errors in speech recognition system via adaptation of models to each other detected speakers. This is shown experimentally for the Turkish Broadcast News data. It is shown that the error rate reduction with using developed baseline automated diarization setup is slightly lower than than using manually made perfect diarization. A second problem in diarization problem is the

determination of the number and identity of speakers in a speech segment in this context. A new speech source decomposition algorithm is presented in solution to this problem. Lastly, a comparative study between factor analysis based audio diarization and the baseline system is analyzed.

1.4. Thesis Organization

In the second chapter of this thesis, common theoretical background for both speaker recognition and diarization is discussed. These include front end processing of the input data, speaker modeling and clustering. In the next chapter, contents only associated with speaker verification problem is discussed. These include some of the speaker verification methodologies used in literature, score normalization, the evaluation metric and lastly experiments associated with speaker verification. In addition to the performance of the baseline systems, microphone suitable systems for NIST SRE 2010 are examined. Audio diarization for Turkish Broadcast News is studied in Chapter 4. Baseline and factor analysis based systems and the proposed approach in overlapping speech diarization and the corresponding experiments are presented.

2. BACKGROUND

In this chapter, the theoretical background of speaker verification and audio diarization will be discussed. Mainly, these systems consist of three main blocks, front-end processing, statistical modeling, scoring. The main blocks speaker verification and audio diarization are illustrated in Figure 2.1 and 2.2, respectively. These main blocks of two systems are the same or similar to each other. For this reason, common theoretical background of these two systems will be discussed in this chapter. First, front end feature vectors are extracted and then speech segments are detected in both systems. In speaker verification task, an audio is expected to contain speech segments from same speaker. However, this is not the case in audio diarization. For this reason, audio diarization contains an additional step in which change points stemming from different audio sources are detected. Also, these change points can be refined in the optimal re-segmentation step of audio diarization. In speaker verification task, statistical modeling and scoring parts are used in the training and test phases, respectively. On the other hand, these blocks are used iteratively in the speaker clustering phase of the audio diarization task.

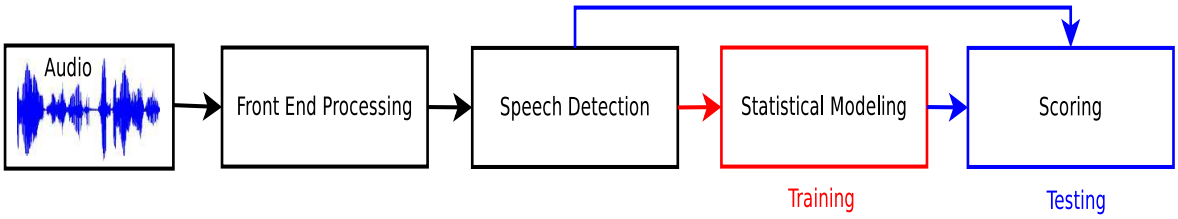

Figure 2.1. The main blocks of speaker verification.

2.1. Front End Processing

First stage of speaker recognition or segmentation is speech parameterization of the input signal on short term spectral content. In the process of speech parameterization, input signal is converted into a vector consisting of features. The purpose of conversion from input signal to a new representation is to have compact, less re-

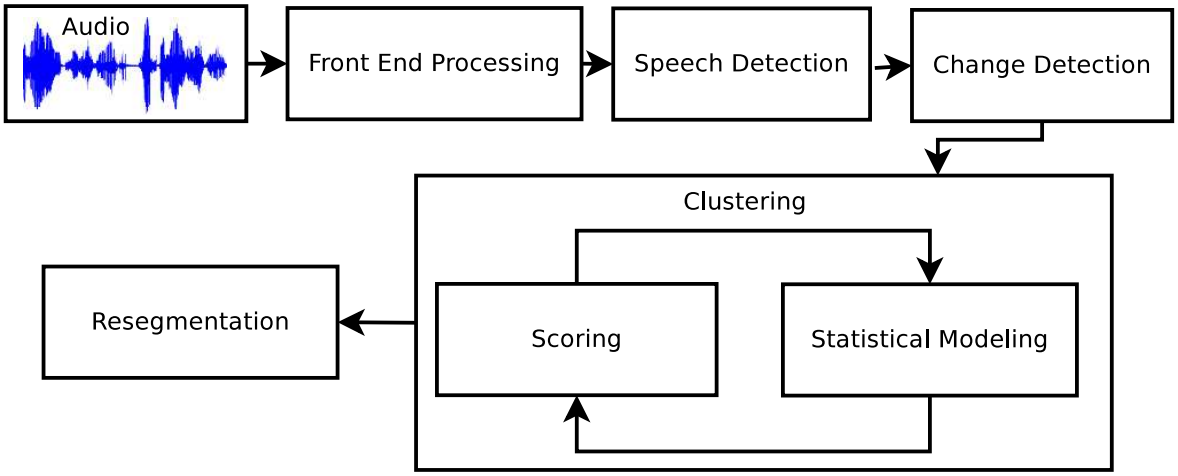

Figure 2.2. The main blocks of audio diarization.

dundant and more suitable representation of speech signal for statistical modeling and scoring calculations. Mel Frequency Cepstral Coefficients (MFCC) [1], Perceptual Linear Predictive Coding (PLP) [2] and Linear Predictive Coding (LPC) [3] are proposed for speech parametrization methods. Most of the systems use MFCC representation and it is used in our studies because Mel-scale and logarithmically spaced filters that is utilized in MFCC are good approximation of human auditory system. MFCC representation was first proposed to be used in Automatic Speech Recognition (ASR) systems and it is supposed to have low inter speaker variation because ASR systems are supposed to be speaker independent. For this reason, it can be expected that efficiency of MFCCs in speaker verification could be low, however it is shown experimentally that it is suitable for speaker recognition task [4].

2.1.1. MFCC Parametrization

Block diagram of MFCC feature extraction is shown in Figure 2.3. The extraction is based on short term spectral features. An utterance is partitioned into overlapping speech segments with generally a 20-30 ms window which are overlapping over 10 ms durations [5]. The speech is first pre-emphasized with a first order difference equation. The aim of this filter is to enhance high frequencies of the speech which is deteriorated in speech production. Hanning and Hamming windows are mostly used for the window

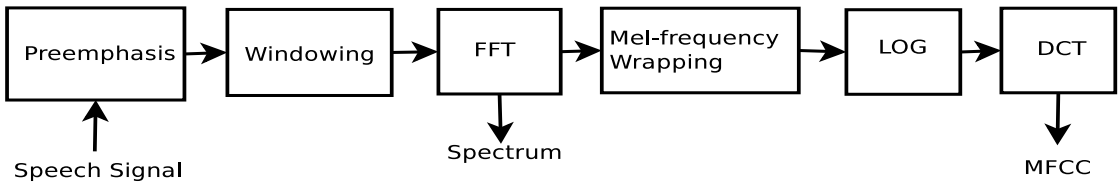

Figure 2.3. MFCC computation scheme.

selection at the windowing step. Short Time Fourier Transform (STFT) is applied to have frequency spectral representation of the waveform. Filterbank coefficients are extracted using triangular bandpass filters spaced according to the mel-scale. Center of the filter are given with the Equation 2.1. These filters are a crude approximation of the human auditory system.

$$f_{mel} = 1000 \frac{\log(1 + f_{lin}/1000)}{\log 2} \tag{2.1}$$

Absolute logarithm of these filterbank coefficients are further processed with Discrete Cosine Transform (DCT) in order to obtain MFCC as given in the Equation 2.2 where f_s are absolute logarithmic spectral coefficients, S is number of spectral coefficients and n is the number of cepstral coefficient to be extracted.

$$c_n = \sum_{s=1}^{S} f_s \cos\left(n\pi(s - 0.5)/S\right) \tag{2.2}$$

First (Δ) and second derivatives ($\Delta\Delta$) of coefficients can be calculated and appended to the vector to increase robustness and overall system performance. These additional coefficients are calculated via linear regression using preceding and subsequent windows. The acceleration coefficients, second derivatives, are calculated over the first derivatives coefficients. It has been experimentally shown that third or higher order derivatives are not effective in speaker recognition performance [6].

2.1.2. Channel Compensation in Front End Vectors

It is common to assume in speaker recognition that feature vectors have statistically independent components. Statistical modeling with these vectors is based on this assumption although it is not always the case. Also, system performance is adversely affected by unwanted channel effects. Several channel compensation techniques are offered in the state of the art speaker recognition systems in order to reduce dependency between features in parametric feature vectors and to reduce intersession variability namely the channel effects.

Cepstral Mean Subtraction (CMS) is the earliest method to reduce channel variations. Slowly varying convolutive channel effects can be eliminated using CMS. Feature vectors are subtracted from the mean vector which is estimated over an utterance or a window. Component-wise variance normalization can also be applied to feature vectors to match unity variance [7].

CMS is able to reduce the impact of slowly varying convolutive noise but when noise is additive, feature estimates degrade significantly [8]. Feature warping [8] can remove both types of noises. The objective is to apply a non-linear transformation so that observation feature vectors will be distributed with normal distribution with zero mean and unit variance. At each analyzed interval containing K samples, feature values are sorted by their values in the K samples separately for each other dimension. The transformed value m for a given feature vector is in Equation 2.3 when feature value is ranked as R^{th} in K samples.

$$\frac{K + 0.5 - R}{K} = \int_{z=-\infty}^{m} \mathcal{N}(z|0,1)\mathrm{d}z \tag{2.3}$$

Here, z is normally distributed random variable with zero mean and unit variance. The window containing K samples, where central vector is to be normalized, is slided one vector at a time. The features are normalized independently of each other.

Another channel compensation is short time gaussianization. Feature warping

assumes that features in a vector are statistically independent. Short time gaussianization weakens this assumption and decorrelates the vectors using a linear transformation. Then, feature warping is performed as a second operation. In this respect, it can be thought that feature warping is a special case of short time gaussianization. Gaussianization is applied on a window containing K samples and operation is applied on the central vector. The linear transformation matrix is estimated on a training set using Expectation Maximization (EM) algorithm. The details of the estimation is explained in [9].

Heteroscedastic Linear Discriminant Analysis (HLDA) [10] is used to decorrelate the feature vector and reduce dimensionality. The difference between HLDA and from Linear Discriminant Analysis (LDA) is that common covariance assumption is relaxed. The classes are determined using alignment of feature vector onto Universal Background Model (UBM), which is discussed in Section 2.2.1.1. Class covariance is estimated to obtain the transformation matrix. Using lower dimensional features new UBM is estimated.

2.2. Speaker Modeling and Classification

After front-end processing, a generative model is trained using front-end vectors to represent a speaker and overall speaker population. GMM is widely used for this purpose. Support Vector Machines, which is a discriminative model, or again generative models like Joint Factor Analysis (JFA) and I-Vectors are used to increase performance of GMM based speaker recognition systems. In the following sections, theory of these generative and discriminative models are given.

2.2.1. Gaussian Mixture Model

One of the most popular form for modeling speech observations is the mixture of Gaussian distributions. The various observations can be considered as examples from different types of phonemes. Also, same phonemes can have different characteristics due to different channel and speaker dependent effects. GMMs can represent approximation

to any distribution using enough number of mixtures. Therefore, it is reasonable to use GMM to represent the underlying distribution of all acoustic features under various channel and speaker dependent conditions.

GMMs are weighted sum combinations of C multivariate Gaussian distributions. The likelihood of an observation $\mathbf{x}$ for given C multivariate Gaussian distribution and its associated parameters of mean μ_c, covariance $\mathbf{\Sigma}_c$ and weight $p(\omega_c)$ is given in Equation 2.4 where θ represents all mean and covariance parameters and ω represents all weight parameters.

$$p(\mathbf{x}|\theta,\omega) = \sum_{c=1}^{C} p(\omega_c)\mathcal{N}(\mathbf{x}|\mu_c,\mathbf{\Sigma}_c) \tag{2.4}$$

This is weighted combination of C Gaussian distributions. Sum of weights, $\sum_{c=1}^{C} p(\omega_j)$, must equal to unity. Likelihood of each multivariate D dimensional Gaussian component density is given in Equation 2.5 for given the observation $\mathbf{x}$.

$$\mathcal{N}(\mathbf{x}|\mu_c,\mathbf{\Sigma}_c) = \frac{1}{(2\pi)^{D/2}|\mathbf{\Sigma}_c|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu_c)^T\mathbf{\Sigma}_c^{-1}(\mathbf{x}-\mu_c)\right\} \tag{2.5}$$

Successive observations are dependent in speech. It is reasonable to expect similar observation in successive observations. However, GMMs represent all possible acoustic and channel effects in a unified model in the speaker recognition and segmentation task. Therefore, each observation is probabilistically aligned with the associated mixture component in an automated fashion when posterior probability of observation that is generated by a mixture component is calculated. This would be impossible if different GMMs are trained for each phoneme. In this case, one needs to resort to statistical models in which first order dependency is preserved between feature variables, like Hidden Markov Models (HMM). Modeling all acoustic contents in a unified model provides independence property between successive observations. Therefore, likelihood

of given K samples $\mathbf{x}_1 \dots \mathbf{x}_K$ is given in Equation 2.6.

$$p(\mathbf{x}_1 \dots \mathbf{x}_K|\theta, \omega) = \prod_{k=1}^{K} \sum_{c=1}^{C} p(\omega_c) p(\mathbf{x}_k|\theta_c) \tag{2.6}$$

Maximum likelihood estimate of the parameters, (ω, θ), are obtained with the parameters that maximize the overall likelihood of the data as given in Equation 2.7.

$$\theta^*, \omega^* = \underset{\theta,\omega}{\operatorname{argmax}} \sum_{k=1}^{K} \log p(\mathbf{x_k}|\theta, \omega) \tag{2.7}$$

The maximum likelihood estimation can be obtained using EM algorithm. In the first step, posterior probabilities of an observation that is generated by all mixture components are calculated given the parameters of the model as in the Equation 2.8. This is repeated for all observations. In the Maximization step, parameters of the model are updated in the direction that maximizes overall likelihood of the data as in the Equation 2.9.

$$p(\omega_c|\mathbf{x_k}, \theta_c) = \frac{p(\omega_c) p(\mathbf{x_k}|\omega_c, \theta_c)}{\sum_{c=1}^{C} p(\omega_c) p(\mathbf{x_k}|\omega_c, \theta_c)} \tag{2.8}$$

$$p(\hat{\omega}_c) = \frac{\sum_{k=1}^{K} p(\omega_c|\mathbf{x_k}, \theta_c)}{K} \qquad \hat{\mu}_c = \frac{\sum_{k=1}^{K} p(\omega_c|\mathbf{x_k}, \theta_c)\mathbf{x}_k}{\sum_{k=1}^{K} p(\omega_c|\mathbf{x_k}, \theta_c)}$$

$$\hat{\mathbf{\Sigma}}_c = \frac{\sum_{k=1}^{K} p(\omega_c|\mathbf{x_k}, \theta_c)(\mathbf{x}_k - \mu_c)(\mathbf{x}_k - \mu_c)^T}{\sum_{k=1}^{K} p(\omega_c|\mathbf{x_k}, \theta_c)} \tag{2.9}$$

It is guaranteed that likelihood of the training data increases monotonically at each iteration. Depending on model complexity and training data, five to ten iterations are generally sufficient for convergence [3]. Details of the parameter estimation can be found at [11].

Covariance parameter of the GMM is generally selected of a diagonal form. It has been shown experimentally that diagonal covariance form of GMMs yield lower error rates in speaker recognition problem than full covariance case [3]. Also, diagonal covariance form of GMM is computationally more efficient because it avoids C matrix inversions in likelihood calculations.

In the following sections, UBM which is basically a GMM which is trained over large speaker population and its adaptation methods to speakers will be discussed.

2.2.1.1. Universal Background Model. A GMM can be trained over various speakers in order to represent a general speaker model and is generally referred to as UBM. This is useful for comparing the likelihood score coming from the speaker model with an inverse hypothesis in the speaker verification task [12]. This approach is used in many earliest GMM based systems. However, UBM is still used in most modern systems in front end processing of the feature vectors.

UBM is supposed to capture speaker independent phonetic feature distributions. Each mixture component is supposed to capture phonetic distributions under various conditions. In this respect, it is utilized to align feature vectors of an utterance with appropriate mixing component and extract speaker statistics for JFA and I-vector analysis which are discussed in the successive sections.

Training data of UBM should be compatible with the one that may be encountered in recognition [13]. For instance, if there is a prior knowledge on gender information of speakers, UBM should be trained with only speakers belonging to that gender. If there is no information on gender, balanced amount of data from both genders should be utilized to avoid biased estimation.

[3] and [14] suggest that training UBM with small amount of data yields similar error rates with the one including extensive amount of data. However, training data for UBM should capture various acoustic and phonetic content in a balanced way. The

study in [14] showed that increasing inter-speaker variability in training data lowered error rates to some extent.

2.2.1.2. Maximum-a-Posteriori Adaptation. Speaker model trained with only the speaker enrollment speech will typically have 64-256 mixture components. Higher order of mixture components will require more data which is not easy to obtain for a single speaker. A better approach is to obtain a speaker model via adaptation of background model parameters. That approach uses speaker's training speech to form Bayesian adaptation or Maximum a Posteriori estimation (MAP) [15]. Using background model's parameters to obtain speaker's model provides tight coupling between the speaker model and the background model. It has been shown experimentally that it has better performance than the decoupled method. Also, this provides a fast scoring technique with possible use of top scoring mixing components [11]. The adaptation has two steps like the EM algorithm. The first step is the same as the expectation step of EM algorithm, but the maximization step differs. The zero, first and second order Baum-Welch sufficient statistics are calculated as in the Equation 2.10 for each other mixture components.

$$\begin{aligned} N_c &= \sum_{k=1}^{K} p(w_c|\mathbf{x}_k, \theta_c) \\ \mathbf{F}_c &= \sum_{k=1}^{K} p(w_c|\mathbf{x}_k, \theta_c)\mathbf{x}_k \\ \mathbf{S}_c &= \text{diag}\left(\sum_{k=1}^{K} p(w_c|\mathbf{x}_k, \theta_c)\mathbf{x}_k\mathbf{x}_k^T\right) \end{aligned} \tag{2.10}$$

$p(w_c|\mathbf{x}_k, \theta_c)$ is the posterior probability of the observation $\mathbf{x}_k$ to be generated by the mixing component c. Speaker adapted parameters are calculated in the Equation 2.11 which are obtained using MAP adaptation for each other mixture components.

$$\begin{aligned} p(\hat{\omega}_c) &= (\alpha_c N_c/K + (1-\alpha)p(\omega_c))\gamma \\ \hat{\mu}_c &= \alpha_c \mathbf{F}_c + (1-\alpha)\mu_c \\ \hat{\mathbf{\Sigma}}_c &= \alpha_c \mathbf{S}_c + (1-\alpha)(\mathbf{\Sigma}_c^2 + \mu_c^2) - \hat{\mu}_c^2 \end{aligned} \tag{2.11}$$

$\omega, \mu, \mathbf{\Sigma}$ are the parameters of UBM. γ is the scaling factor ensuring that sum of weights equals to unity. The adaptation coefficient which controls the balance between old and new estimates is given in Equation 2.12.

$$\alpha_c = \frac{N_c}{N_c + r} \tag{2.12}$$

When adaptation data has low probability for the mixing component c, α_c becomes low. Estimated parameters, for that case, are more affected by UBM's parameters than user statistics and less affected by UBM's parameters when probability of adaptation data is high.

Experimentally, it has been shown in NIST evaluations that adaptation of covariance matrix and weights increased error rates. For this reason it is common to adapt only mean vectors and copy both covariance and weights parameters of UBM to construct a speaker model.

2.2.1.3. Maximum Likelihood Linear Regression. Another adaptation method is Maximum Likelihood Linear Regression (MLLR), that is generally used for speaker adaptation of speaker independent acoustic models of speech recognition. The method is to estimate the adaptation matrix $\mathbf{Y}$ through maximum likelihood and then apply the linear transformation. Adaptation is performed on only mean values for similar reasons as in the MAP case. It is assumed that the difference between speakers are due to mean vectors [16].

The adaptation matrix is of the form given in Equation 2.13 for the state s. φ is the offset term.

$$\xi = [\varphi, \mu]^T \qquad \hat{\mu}_s = \mathbf{Y}_s \xi_s \tag{2.13}$$

Maximum Likelihood estimation of $\mathbf{Y}_s$ is given in Equation 2.14.

$$\sum_{k=1}^{K} p(s|\mathbf{x}_k, \theta_s)\mathbf{\Sigma}_s^{-1}\mathbf{x}_k\xi_s^T = \sum_{k=1}^{K} p(s|\mathbf{x}_k, \theta_s)\mathbf{\Sigma}_s^{-1}\hat{\mathbf{Y}}_s\xi_s\xi_s^T \tag{2.14}$$

Here, states are represented with multivariate Gaussian distributions. When states are represented with C mixture GMMs, the adaptation matrix $\mathbf{Y}_{s,c}$ can be estimated separately for each mixture components.

Equation 2.14 is simplified into Equation 2.15 when the covariance matrix is not updated and every frame is aligned to one state through Viterbi decoding. $\delta_{s,k}$ is one when frame k is aligned with the state s and zero otherwise.

$$\sum_{k=1}^{K} \mathbf{x}_k\xi_s^T\delta_{s,k} = \hat{\mathbf{Y}}_s \sum_{k=1}^{K} \xi_s\xi_s^T\delta_{s,k} \tag{2.15}$$

MLLR can be applied to GMM with larger number of mixture components and all parameters are updated even with small amount of data. Iterative EM algorithm can be applied to obtain model parameters. In the first E-step, posterior probabilities are calculated using forward backward or Viterbi algorithm. In the M-step, parameters are calculated as in Equation 2.14 for forward backward case and as in Equation 2.15 for the Viterbi case to maximize the overall probability of all training data.

2.2.2. Support Vector Machines

It has been shown over the last two decades that the SVM, which is a binary discriminative classifier, performed good on a wide variety of machine learning applications. The aim in SVM is to find the optimal separating hyperplane which maximizes the margin between closest training examples of the both classes to the decision boundary. This hyperplane is referred as maximum-margin decision boundary and it minimizes the overall classification error. Support vectors are the ones in or on this margin. A typical example of a SVM is illustrated in Figure 2.4 where linear decision

boundary is of the form as in Equation 2.20.

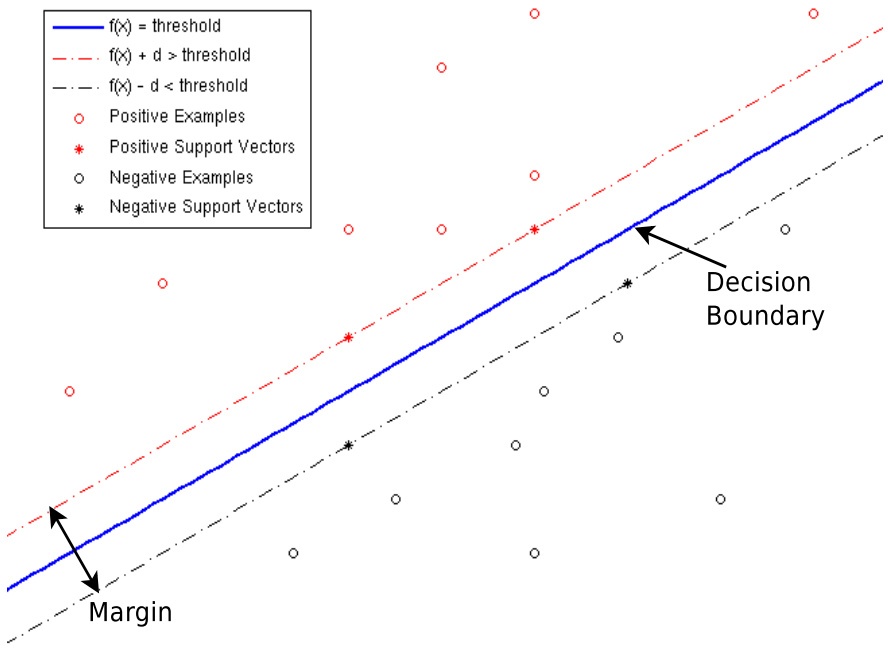

Figure 2.4. A typical example of a SVM.

Mathematical form of the two class classifier SVM is and given in Equation 2.16,

$$f(\mathbf{x}) = \sum_{k=1}^{K} \alpha_k t_k K(\mathbf{x}, \mathbf{x}_k) + d \tag{2.16}$$

Here, t_k are the ideal outputs, $t_k \in \{1, -1\}$. t_k is 1 when $\mathbf{x}_k$ is from positive class and -1 when it is from negative class. d is learned constant parameter. Also, there is constraint that $\sum_{k=1}^{K} \alpha_k t_k = 0$. The α_k and set of $\mathbf{x}_k$ are learned through quadratic optimization using the training set. Optimization is based on maximum margin concept. The purpose of SVM training is to determine optimal boundary between classes. The data lying in the boundaries in Equation 2.16 are the support vectors. $\mathbf{x}_k$ with $\alpha_k > 0$ are used as support vectors and all other $\mathbf{x}_k$ with $\alpha_k = 0$ are not used in testing. For the classification purpose, $f(\mathbf{x})$ is used with an appropriate

threshold.

Mercer condition is required in SVM optimization in order to ensure validity of the margin concept and SVM optimization to be bounded. The kernel is required to be positive semi-definite due to Mercer condition. $K(\mathbf{x}, \mathbf{y})$ can be written as $b(\mathbf{x})^T b(\mathbf{y})$. $b(\mathbf{x})$ is the mapping from the input space [17].

Support vector machines are applied to GMM based Speaker verification task at feature level [18] and model level [17] and score level [19]. The model level approach gained more interest and it will be discussed here.

Given an utterance, MAP adaptation of only means is performed to have speaker GMMs. GMM supervector is obtained by stacking means of a GMM. This can be thought as mapping an utterance to a new high dimensional space. This forms the bases for SVM kernel. SVM kernel compares two utterances u_a and u_b directly using $K(u_a, u_b)$. $K(u_a, u_b)$ can be written as $b(u_a)^T b(u_b)$ thanks to Mercer condition.

2.2.2.1. GMM Supervector Linear Kernel. One common approach in constructing a kernel is based on the use of Kullback Leibler (KL) distance between two utterances, u_a and u_b It has outperforming performance among the other kernels [17] and for this reason only it will be discussed here. Speaker models are $p_a(\mathbf{x}|\theta_a)$ and $p_b(\mathbf{x}|\theta_b)$ trained as GMMs for u_a and u_b respectively. Then, KL distance between these these distributions is defined as in Equation 2.17.

$$D(p_a(\mathbf{x}|\theta_a)||p_b(\mathbf{x}|\theta_b)) = \int_{R^n} p_a(\mathbf{x}|\theta_a) \log\left(\frac{p_a(\mathbf{x}|\theta_a)}{p_b(\mathbf{x}|\theta_b)}\right) d\mathbf{x} \tag{2.17}$$

However this distance cannot be used as a kernel because it violates Mercer condition, it is not symmetric in the first place. Instead of using the divergence in Equation 2.17, upper bound approximation via log sum inequality is used as in Equation 2.18.

$$D(p_a(\mathbf{x}|\theta_a)||p_b(\mathbf{x}|\theta_b)) < \sum_{c=1}^{C} p(\omega_c) D(\mathcal{N}(\mathbf{x}|\mu_c^a, \mathbf{\Sigma}_c)||\mathcal{N}(\mathbf{x}|\mu_c^b, \mathbf{\Sigma}_c)) \tag{2.18}$$

Divergence between two Gaussian distribution in Equation 2.18 with diagonal covariance matrices $\mathbf{\Sigma}_c$ is simplified into the Equation 2.19.

$$\sum_{c=1}^{C} p(\omega_c)(\mu_c^a)^T \mathbf{\Sigma}_c^{-1}(\mu_c^b) \tag{2.19}$$

Because the kernel in Equation 2.19 is a linear dot product, SVM has a compact representation as in Equation 2.20.

$$f(\mathbf{x}) = \left(\sum_{k=1}^{K} \alpha_k t_k b(\mathbf{x}_k) \right)^T b(\mathbf{x}) + d = \mathbf{v}^T b(\mathbf{x}) + d \tag{2.20}$$

It is an advantage to use this kernel because one needs to store only the weighted sum of support vectors. Then, it will be sufficient to use a single inner product to obtain score from SVM classification. The score obtained in SVM is not however a probabilistic score. In order to obtain a probabilistic score, one can consider use of a sigmoid function.

2.2.3. Joint Factor Analysis

Performance of speaker verification systems are susceptible to unwanted variations in the speech signals. These variations are mainly caused by speaker dependent and speaker independent effects. Speaker dependent variations are the variations among session recordings and speaker independent variations are caused by channel effects such as different microphone usage among sessions or environmental effects such as different background noises. There has been efforts to compensate these variations to improve performance of speaker verification task. JFA [20, 21] is one of the most successful approaches to model both speaker and channel variability. It is both applied to cepstral features and continuous prosodic features as in [22]. JFA is based on GMM structured speaker verification system. It is assumed that mean supervector of the GMM is the sum of session and speaker variabilities which are distributed in lower dimensional speaker and channel factors. JFA is similar to feature mapping approach in [23] but it differs in assumption that channel variabilities are continuous whereas

the latter one assumes discrete variability. All latent random variabilities are assumed to be normally distributed with zero mean and unit variances.

The structure of the JFA can be thought as a single prior containing classical, eigenvoice and eigenchannel MAP. GMM structure consists of C mixture Gaussian and D dimensional acoustic feature vectors. Speaker and session dependent observation supervector is the sum of speaker and channel-dependent supervectors which are statistically independent and normally distributed. An utterance spoken by speaker s during session k, the observation is normally distributed with mean $\mathbf{M}_k(s)$ and with a covariance $\mathbf{\Sigma}$. $\mathbf{M}_k(s)$ is the sum of speaker dependent supervector $\mathbf{M}(s)$ and channel variation supervectors as given in Equation 2.21.

$$\mathbf{M}_k(s) = \mathbf{M}(s) + \mathbf{U}\mathbf{x}_k \tag{2.21}$$

$\mathbf{U}$ is the low rank channel factors loading matrix which captures session dependent channel variation in a subspace and $\mathbf{x}_k$ is the session dependent latent variable. The session independent component $\mathbf{M}(s)$ is decomposed as a speaker independent mean supervector $\mathbf{m}$ and two speaker dependent supervectors as given in Equation 2.22.

$$\mathbf{M}(s) = \mathbf{m} + \mathbf{V}\mathbf{y}(s) + \mathbf{D}\mathbf{z}(s) \tag{2.22}$$

Speaker independent supervector $\mathbf{m}$ can be taken as the mean supervector of UBM. $\mathbf{V}$ is the low rank speaker factors loading matrix capturing speaker factors. In order to capture residual speaker factors that is not modeled by $\mathbf{y}(s)$, a diagonal factor loading matrix $\mathbf{D}$ can be added to the JFA structure. However, latent factors $\mathbf{z}(s)$ whose dimension is same as supervector's dimension increases complexity of the problem. It is common not to include $\mathbf{D}$ to avoid dramatic increase in the complexity. Classical MAP, eigenvoice MAP [24] and eigenchannel MAP [25] is the case when there is only $\mathbf{D}$, $\mathbf{V}$ and $\mathbf{U}$ term in the JFA structure, respectively. The JFA model can be described by quintuple Λ which is of the form $(\mathbf{m}, \mathbf{U}, \mathbf{V}, \mathbf{D}, \mathbf{\Sigma})$.

The likelihood calculations of JFA is as follows. Observations consisting of K utterances associated with speaker s is denoted as $o(s)$. The total latent variable is in Equation 2.23.

$$\underline{\mathbf{X}} = [\mathbf{x}_1(s)^T, \ldots, \mathbf{x}_K(s)^T, \mathbf{y}(s)^T, \mathbf{z}(s)^T] \tag{2.23}$$

If the latent variables are known, $\mathbf{M}_k(s)$ and calculation of likelihood of the observation would be possible. One needs to marginalize over the latent variables to calculate likelihood of the observations as in Equation 2.24.

$$p(o(s)|\Lambda) = \int p(o(s)|\underline{\mathbf{X}}, \Lambda)\mathcal{N}(\underline{\mathbf{X}}|\mathbf{0}, \mathbf{I})d\underline{\mathbf{X}} \tag{2.24}$$

Solution to this integral is given in Equation 2.25 where c is the corresponding mixture component of parameters.

$$\begin{aligned}
\log p(o(s)|\Lambda) &= \frac{1}{2}\log|\mathbf{\Sigma}| + \frac{1}{2}\sum_{k=1}^{K}(E[\mathbf{x}_k^T]\sum_{c=1}^{C}\mathbf{U}_c^T\mathbf{\Sigma}_c^{-1}\overline{\mathbf{F}}_c^k \\
&+ E[\mathbf{y}(s)^T]\sum_{c=1}^{C}\mathbf{V}_c^T\mathbf{\Sigma}_c^{-1}\overline{\mathbf{F}}_c^k + \sum_{c=1}^{C}E[\mathbf{z}(s)_c^T]\mathbf{D}_c\mathbf{\Sigma}_c^{-1}\overline{\mathbf{F}}_c^k \\
&+ \sum_{k=1}^{K}\sum_{c=1}^{C}\left(N_c^k\log\frac{1}{(2\pi)^{\frac{D}{2}}|\mathbf{\Sigma}_c|^{0.5}} - \frac{1}{2}\mathrm{tr}(\mathbf{\Sigma}_c^{-1}\overline{\mathbf{S}}_c^k)\right)
\end{aligned} \tag{2.25}$$

The expected values of the latent variables and proof of the likelihood is studied in [26]. N_c is zero, $\overline{\mathbf{F}}_c$, $\overline{\mathbf{S}}_c$ are mean vector ($\mathbf{m}$) removed first and second order Baum-Welch statistics for mixture c in Equation 2.10.

It is only possible to calculate joint likelihood of the utterance from a speaker as in Equation 2.25 because speaker dependent factors are same for a speaker. Given a test utterance o^t, log-likelihood ratio can be calculated by comparing joint likelihoods of enrollment and test data for same speaker hypothesis and independent likelihoods of test and enrollment data for different speaker hypothesis as given in the Equation

2.26.

$$\log \frac{p(o(s), o^t|\Lambda)}{p(o(s)|\Lambda)p(o^t|\Lambda)} \tag{2.26}$$

However, computation of the logarithmic likelihood in Equation 2.26 is computationally infeasible because expected values of all terms needs to be estimated. Instead of this, parameters of JFA, Λ can be updated to generate likelihood ratio as given in Equation 2.27.

$$\log \frac{p(o^t|\Lambda(s))}{p(o^t|\Lambda)} \tag{2.27}$$

The updated parameters are usually $\mathbf{D}$, $\mathbf{V}$ for the enrolled speaker and $\mathbf{U}$ and $\mathbf{\Sigma}$ is not updated because these session dependent parameters are independent of a speaker.

Parameter estimation of JFA aims to maximize total likelihood of the all training data which is in Equation 2.28.

$$\sum_{s=1}^{S} \log p(o(s)|\Lambda) \tag{2.28}$$

Parameters of JFA, Λ can be estimated via EM algorithm which guarantees increase in total likelihood at each iterations. Two estimation algorithms for parameters of JFA are Maximum Likelihood approach which is discussed in [24] and Minimum Divergence approach which is discussed in [27]. The latter one converges faster but it keeps orientation of speaker and channel factors constant so it is better to use minimum divergence after maximum likelihood estimation is once performed. Therefore, generally all parameters are estimated with maximum likelihood and then speaker adaptation is performed using minimum divergence to adapt the parameters $\mathbf{V}$ and $\mathbf{D}$.

2.2.4. I-Vector Analysis

In the study [28], it is observed that channel variability matrix in the JFA model captures some speaker variabilities as well. In [29], a new feature extractor based on factor analysis is offered in which speaker and channel variabilities are captured through one low dimension subspace. The low dimensional space in which features lie is named as total variability space. Speaker and channel dependent supervector is given in Equation 2.29.

$$\mathbf{M} = \mathbf{m} + \mathbf{T}\mathbf{w} \tag{2.29}$$

The supervector $\mathbf{M}$ has a mean $\mathbf{m}$ and a covariance $\mathbf{\Sigma}$ and these parameters are usually obtained from the mean and covariance parameters of the UBM. $\mathbf{T}$ is total variability matrix and $\mathbf{w}$ is the normally distributed latent variable with mean $\mathbf{w}_o$ and covariance $\mathbf{L}_o^{-1}$. The posterior distribution of $\mathbf{w}$ is given as in Equation 2.30.

$$p(\mathbf{w}|o(s)) = \mathcal{N}(\mathbf{w}|\mathbf{w}_o, \mathbf{L}_o^{-1}) \tag{2.30}$$

The MAP point estimate of the posterior distribution which is equal to the mean, $\mathbf{w}_o$ is referred as I-vector [30] and the $\mathbf{T}$ matrix is referred as the I-vector extractor.

First and zero order statistics given in Equation 2.10 are used as input data. $\mathbf{T}_c$ is the cth component of total C mixtures of total variability matrix.

The point estimate of the latent variable, I-vector, for a given observation data is given in Equation 2.31 whose detailed study can be found in [24].

$$\mathbf{w}_o = \mathbf{L}_o^{-1}\mathbf{T}^T\mathbf{\Sigma}^{-1}\overline{\mathbf{F}} \tag{2.31}$$

The supervector $\overline{\mathbf{F}}$ is obtained by concatenating $\overline{\mathbf{F}}_c$ vectors ($\overline{\mathbf{F}} = [\overline{\mathbf{F}}_1; \ldots; \overline{\mathbf{F}}_C]$). The covariance matrix of the random variable $\mathbf{w}$ is given as $\mathbf{L}^{-1}$. It is computed as in

Equation 2.32.

$$\mathbf{L}_o = \mathbf{I} + \sum_{c=1}^{C} N_c \mathbf{T}_c^T \mathbf{\Sigma}^{-1} \mathbf{T}_c \tag{2.32}$$

Parameter estimation of the total variability matrix is the same as estimation of eigenvoice matrix as in [24] but it differs in one respect. In the latter one, utterances from same speakers are considered to be produced by the same speakers but the former one assumes that they are produced by different speakers. Parameters are estimated through EM algorithm which is the same as the JFA case. In the expectation step accumulators are calculated over all K training utterances and for all C mixture components as in Equation 2.33.

$$\begin{aligned} \overline{\mathbf{C}} &= \sum_{k=1}^{K} \overline{\mathbf{F}}^k \mathbf{w}_k^T \\ \overline{\mathbf{A}}_c &= \sum_{k=1}^{K} N_c^k (\mathbf{L}_k^{-1} + \mathbf{w}_k \mathbf{w}_k^T) \end{aligned} \tag{2.33}$$

In the Maximization step update Equation 2.34, new estimate of the total variability matrix is calculated.

$$\overline{\mathbf{T}}_c = \overline{\mathbf{C}\mathbf{A}}_c^{-1} \tag{2.34}$$

Estimates of $\mathbf{w}$ and $\mathbf{L}$ is obtained by Equation 2.31 and 2.32, respectively. Additionally, minimum divergence step in [31] can be added for quicker convergence. The update step is in the Equation 2.37 where the operation $-\frac{1}{2}$ is Cholesky decomposition of a matrix. $\overline{\mathbf{A}}_{md}$ and $\overline{\mathbf{C}}_{md}$ are the collected accumulators as in Equation 2.37.

$$\overline{\mathbf{C}}_{md} = \frac{1}{K} \sum_{k=1}^{K} \mathbf{w}_k^T \tag{2.35}$$

$$\overline{\mathbf{A}}_{md} = \frac{1}{K} \sum_{k=1}^{K} N_c^k (\mathbf{L}_k^{-1} + \mathbf{w}_k \mathbf{w}_k^T) \tag{2.36}$$

$$\hat{\mathbf{T}} = \left(\overline{\mathbf{A}}_{md} - \overline{\mathbf{C}}_{md} \overline{\mathbf{C}}_{md}^T \right)^{-\frac{1}{2}} \mathbf{T} \tag{2.37}$$

2.2.4.1. Scoring Methods for I-Vectors. In [30], two scoring methods, SVM and Cosine distance, for I-Vector analysis is studied. In the first one, I-vectors are used as input vectors to SVM. Best results with SVM are obtained with cosine distance kernel out of linear and Gaussian kernels [32]. The cosine kernel is given in Equation 2.38.

$$k(\mathbf{w}_1, \mathbf{w}_2) = \frac{\langle \mathbf{w}_1, \mathbf{w}_2 \rangle}{\|\mathbf{w}_1\| \|\mathbf{w}_2\|} \tag{2.38}$$

Magnitude of the vectors are affected mostly by channel and session variations. To reduce this effect, I-Vectors are normalized by their norms. This increases the robustness of the system because it is believed that the difference among different speakers are caused by the angle between I-vectors.

Another scoring method is direct use of cosine distance value between target and test I-vectors as w_1 and w_2 vectors in Equation 2.38. In this method, there is no target enrollment as JFA, SVM and GMM/UBM systems. Both training and testing I-vectors are extract in same way. Cosine distance score is generated by a single inner product. This enables faster calculation of computationally expensive score normalization techniques such as t-norms and z-norms.

Within Class Covariance Normalization (WCCN) [33], LDA [35] and Nuisance Attribute Projection (NAP) [34] are three channel compensation techniques to remove nuisance effects. Thanks to lower dimensionality, application of these techniques in total variability space provides faster computation than other compensation techniques applied on the GMM supervector space.

The purpose of WCCN in [33] is to minimize expected total error rate which includes false rejection and acceptance in the SVM training. The solution to this problem is a generalized linear kernel as in Equation 2.39.

$$k(\mathbf{w}_1, \mathbf{w}_2) = \mathbf{w}_1^T \mathbf{W}^{-1} \mathbf{w}_2 \tag{2.39}$$

$\mathbf{W}$ is the within class covariance matrix. Each utterance of a speaker is considered as

within class member and $\mathbf{W}$ is computed as in Equation 2.40.

$$\mathbf{W} = \frac{1}{S}\sum_{s=1}^{S}\frac{1}{K_s}\sum_{k=1}^{K_s}(\mathbf{w}_k^s - \overline{\mathbf{w}}_s)(\mathbf{w}_k^s - \overline{\mathbf{w}}_s)^T \tag{2.40}$$

K_s is the number of utterances for the speaker s and $\overline{\mathbf{w}}_s$ is mean I-vector computed over all utterances of speaker s. [30] uses Cholesky decomposition of matrix $\mathbf{W}$ as $\mathbf{W}^{-1} = \mathbf{B}\mathbf{B}^T$ to preserve cosine distance property as given in Equation 2.41.

$$k(\mathbf{w}_1, \mathbf{w}_2) = \frac{(\mathbf{B}^T\mathbf{w}_1)^T(\mathbf{B}^T\mathbf{w}_2)}{\sqrt{(\mathbf{B}^T\mathbf{w}_1)^T(\mathbf{B}^T\mathbf{w}_1)}\sqrt{(\mathbf{B}^T\mathbf{w}_2)^T(\mathbf{B}^T\mathbf{w}_2)}} \tag{2.41}$$

The purpose of LDA is to project I-vectors onto a lower dimensional space where intraclass variance is minimized and interclass variance is maximized. Same speaker enrollments are member of intra-class and variance is mostly caused by channel effects. On the other hand, inter-class variance is caused by different speakers. In this approach, better discrimination between different speakers is aimed using new orthogonal axes. A low rank projection matrix is estimated through the optimization of Rayleigh coefficient as in Equation 16 of [30]. Instead of using full rank $\mathbf{B}$ matrix as in Equation 2.41, score is generated using lower dimensional projection matrix $\mathbf{A}$, which is obtained via LDA training, as in Equation 2.41. A better approach is to use LDA and WCCN both simultaneously on I-vectors to generate score and it is referred as cosine scoring with WCCN/LDA. The WCCN/LDA solution is given in Equation 2.42.

$$k(\mathbf{w}_1, \mathbf{w}_2) = \frac{(\mathbf{A}^T\mathbf{w}_1)^T\mathbf{W}^{-1}(\mathbf{A}^T\mathbf{w}_2)}{\sqrt{(\mathbf{A}^T\mathbf{w}_1)^T\mathbf{W}^{-1}(\mathbf{A}^T\mathbf{w}_1)}\sqrt{(\mathbf{A}^T\mathbf{w}_2)^T\mathbf{W}^{-1}(\mathbf{A}^T\mathbf{w}_2)}} \tag{2.42}$$

In NAP method, it is aimed to remove nuisance directions. The channel space, $\mathbf{R}$, is assumed to lie in the eigenvectors with highest eigenvalues of within class covariance because most of the within class variations are assumed to be caused by channel effects. The projection matrix is based on orthogonal complementary space of channel space

which is speaker space and it is given in Equation 2.43.

$$\mathbf{P} = \mathbf{I} - \mathbf{R}\mathbf{R}^T \tag{2.43}$$

The cosine score or kernel for this can be obtained after projecting I-vectors onto this new space as in Equation 2.41 for generation of direct score or SVM modeling of I-vectors, respectively.

2.2.4.2. Probabilistic Linear Discriminant Analysis. Use of Probabilistic Linear Discriminant Analysis (PLDA) in speaker recognition problem is offered in [36]. It is a generative model to explain channel and speaker factors like JFA but it differs in some respects. PLDA can be thought as a JFA model in which there is a single multivariate Gaussian supervector. I-vectors are considered as the features in PLDA training whereas JFA is modeled at the GMM supervector level. Total variability space, which has a lower dimensionality than supervector space, enables full Bayesian treatment of model parameters. It has been shown that use of heavily tailed priors instead of Gaussian in PLDA modeling lowers unfavorable effects of outliers data, thus resulting in lower error rates in speaker recognition task on telephone data.

For a given speaker and K utterances of the speaker, the related I-vector decomposition assumption in [36] is given in Equation 2.44.

$$\mathbf{w}_k = \mathbf{S} + \mathbf{C}_k \tag{2.44}$$

For the Gaussian prior modeling, it is assumed that both speaker dependent component $\mathbf{S}$ and channel dependent component $\mathbf{C}_k$ of I-vector are statistically independent multivariate normally distributed random variables. In heavily tailed prior assumption, these assumptions are relaxed. The formulation of Gaussian PLDA is given in Equation 2.45 where hidden variable $\mathbf{x}_1$ depends on speaker factors and $\mathbf{x}_{2k}$ depends

on channel factors which varies over all recordings of a speaker.

$$\mathbf{w}_k = \mathbf{m} + \mathbf{U}_1\mathbf{x}_1 + \mathbf{U}_2\mathbf{x}_{2k} + \epsilon_k \tag{2.45}$$

The columns of $\mathbf{U}_1$ are the eigenvoices and the columns of $\mathbf{U}_2$ are the eigenchannels. $\mathbf{m}$ is the center of the acoustic space. The residual variability is captured by ϵ_k which is normally distributed with zero mean and covariance $\mathbf{\Lambda}$ which is usually diagonal. The graphical model of the Gaussian prior is given in the following Figure 2.5.

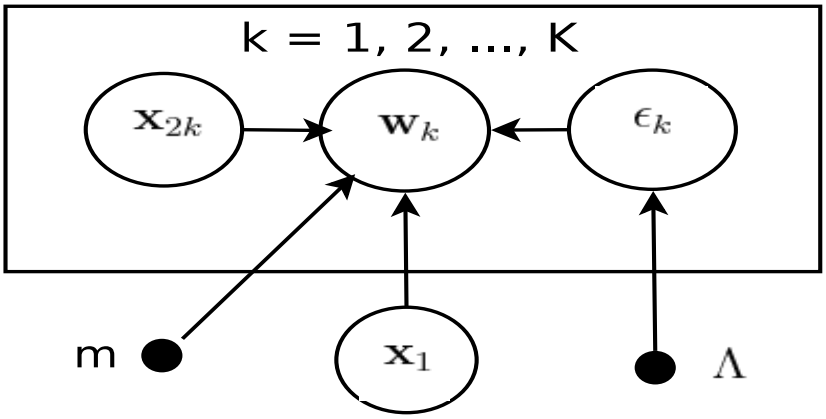

Figure 2.5. Graphical model of Gaussian PLDA.

It can be deduced from the Equation 2.44 and 2.45 that $\mathbf{S} = \mathbf{m} + \mathbf{U}_1\mathbf{x}_1$ and $Cov(\mathbf{S}, \mathbf{S}) = \mathbf{U}_1\mathbf{U}_1^T$ and similarly $\mathbf{C}_k = \mathbf{U}_2\mathbf{x}_{2k} + \epsilon_k$ and $Cov(\mathbf{C}_k, \mathbf{C}_k) = \mathbf{U}_2\mathbf{U}_2^T + \mathbf{\Lambda}^{-1}$. In the heavily tailed prior assumption, all hidden variables are associated with a student-t distribution. Given all I-vector for a speaker, the marginal likelihood is obtained through integration over all hidden variables as in the Equation 2.24. The conditional posterior distribution of the hidden variabilities are obtained using proper variational Bayesian methods [36].

Given two I-vectors $\mathbf{w}_1$ and $\mathbf{w}_2$, the scoring scheme is given in the Equation 2.46 which is a likelihood ratio.

$$\frac{p(\mathbf{w}_1, \mathbf{w}_2|H_1)}{p(\mathbf{w}_1, H_0)p(\mathbf{w}_2, H_0)} \tag{2.46}$$

Here, H_1 is the hypothesis that $\mathbf{w}_1$ and $\mathbf{w}_2$ are the I-vectors of same speaker and H_0 is the hypothesis that they belong to different speakers. Again every term here is the posterior distribution of the hidden variabilities as in Equation 2.24. The approximate likelihoods can be deduced from the Figure 2.5. For the numerator in Equation 2.46, K equals to two which are $\mathbf{w}_1$ and $\mathbf{w}_2$ in the Figure 2.5 and denominator both I-vector likelihoods are calculated separately.

Two parameter estimations methods, Maximum Likelihood and Minimum Divergence, are offered in [36] which are similar to the JFA case except that there is a single Gaussian supervector assumption for the Gaussian prior case. Maximization Likelihood parameter estimation tries to maximize likelihood of all given I-vector data and details of estimation can be found in [37] and full Bayesian treatment can be found in [38].

3. SPEAKER VERIFICATION

Speaker verification is a binary classification task. The aim is to determine whether a spoken utterance belongs to the given speaker identity by using corresponding training enrollment data. There are two types of speaker verification systems, text dependent and independent. Text dependent systems require correct transcription of the utterances. Most of the research and NIST SRE as well as this work are on text independent systems. In the following sections of this chapter, overview of speaker verification, three different text independent speaker verification setups, score normalization, evaluation metric and experiments are presented.

3.1. Overview of a Speaker Verification System

A typical speaker verification system consisting of training and testing phases is illustrated in the Figure 3.1. The first step is front end processing of speech waveform of training and test enrollments.

The details of feature extraction is discussed in Section 2.1.2. Second step is the probabilistic or discriminative modeling of speaker enrollment. Three different GMM based speaker verification methods are discussed in the following sections. These are GMM/UBM, GMM/SVM and factor analysis based verification systems. Lastly, a score is generated for the binary classification task between claimed enrollment speaker and test utterance. Binary decision is made using the predefined optimal threshold in the sense that it minimizes the binary classification error. Optionally, score normalization is applied on the output score to increase robustness of the system to intersession and intra session variabilities.

3.2. Speaker Recognition Methodologies

Three different speaker recognition methodologies are discussed in this section. The first two of them are classical GMM/UBM and GMM/SVM systems. The last

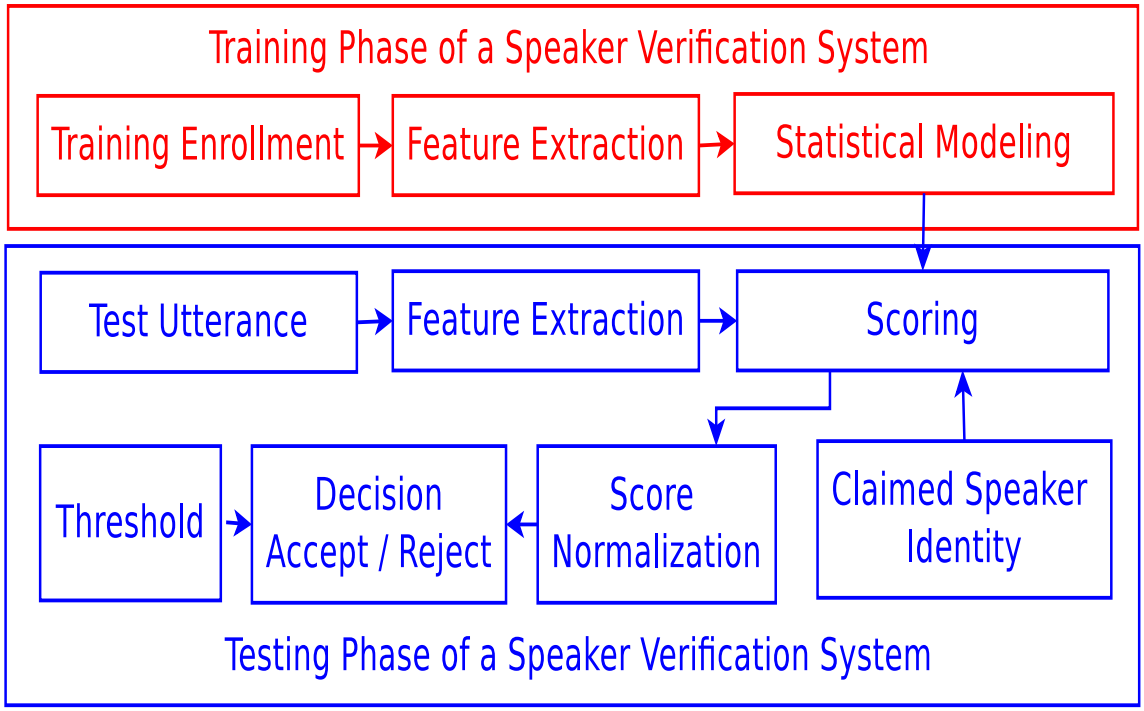

Figure 3.1. Block diagram of speaker verification system.

one is the factor analysis based speaker verification system and a variant of this system which is specialized on microphone conditions in NIST SREs.

3.2.1. GMM/UBM based Speaker Verification System

First successful speaker verification systems used a GMM based generative model. The generative model assigns a likelihood to speech data. In GMM/UBM methodology, two likelihoods coming from two generative models, speaker specific and general speaker models are used in the decision phase. The general speaker model, UBM is estimated through EM algorithm using large amounts of data and typically is composed of 1024 or 2048 mixture components. The speaker model is estimated through MAP of UBM using speaker enrollment data. The scheme of GMM/UBM system is depicted in Figure 3.2.

The logarithmic likelihood ratio is given Equation 3.1. H_1 is the hypothesis that the utterance $\mathbf{X}$ belongs to the claimed identity and H_0 is the opposite hypothesis.

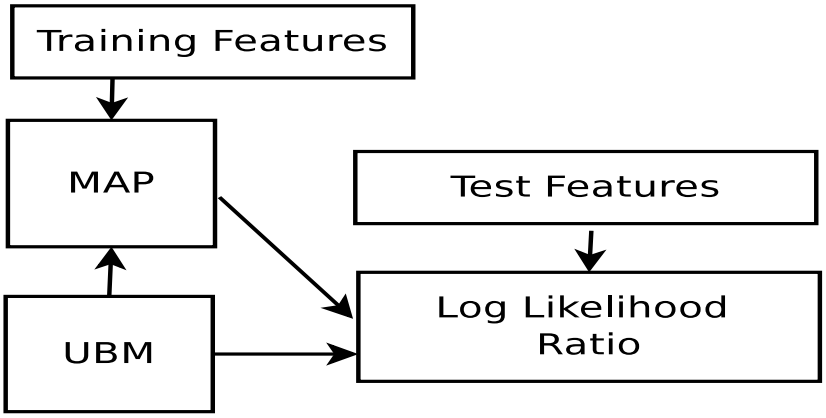

Figure 3.2. Block diagram of GMM/UBM system.

Numerator and denominator of Equation 3.1 are likelihood scores of the data $\mathbf{X}$ given speaker adapted MAP estimation of the speaker GMM and UBM for H_1 and H_0 hypothesis, respectively. Binary decision is made using the threshold θ.

$$\log \frac{p(\mathbf{X}|H_1)}{p(\mathbf{X}|H_0)} \lessgtr \theta \tag{3.1}$$

3.2.2. GMM/SVM based Speaker Verification System

Discriminative speaker verification systems gained interest because they outperformed GMM/UBM systems [17,18]. The GMM/SVM system on model level proposed in [17] is depicted in Figure 3.3.

The parameters of SVM in Equation 2.20 is trained for each speaker. Positive examples of training data are obtained with the speaker enrollment data and negative examples are obtained with large number of different background speaker data. There is an imbalance between positive and negative data because it is not easy to collect sufficient number of speaker enrollments whereas imposter data is abundant. Especially, there is only a single utterance as enrollment data in NIST SRE core test. One possible solution to produce more than one positive example is partitioning of enrollment data as described in [39]. GMM Supervector is obtained by stacking means

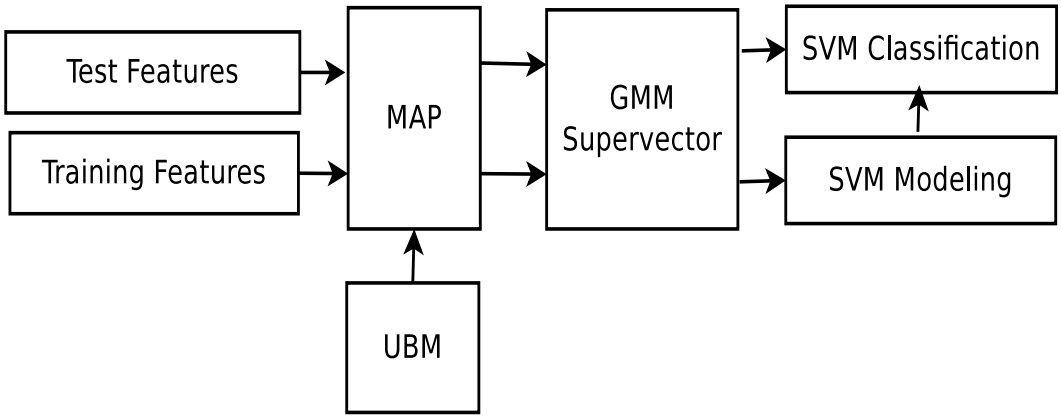

Figure 3.3. Block diagram of GMM/SVM system.

of MAP estimates of speaker enrollments. Using GMM Supervector Linear Kernel as described in Section 2.2.2.1, score is obtained with a single inner product. NAP and WCCN techniques are used to compensate possible channel effects.

3.2.3. Factor Analysis based Speaker Verification System

Factor analysis based speaker verification systems, either JFA or I-vector approach assume that there are hidden channel and speaker factors which collectively generate the observation supervector. They gained interest in the last year because they provide state-of-the-art performance in speaker verification task [30]. Factor analysis based speaker verification system consists of two phases as described in the previous chapter. In the training phase, factor loading matrices are estimated through maximum likelihood or minimum divergence. In test phase, likelihoods of hypothesis are calculated using the generative models.

Training the total variability matrix for I-vector extractor with insufficient amount of microphone data was a problem in NIST SRE 2010. A solution to this problem is offered in the study [40] and depicted in Figure 3.4.

In this model, first the total variability matrix of rank R_{tel} is trained using large amounts of telephone data. First order statistics are centered around the estimated

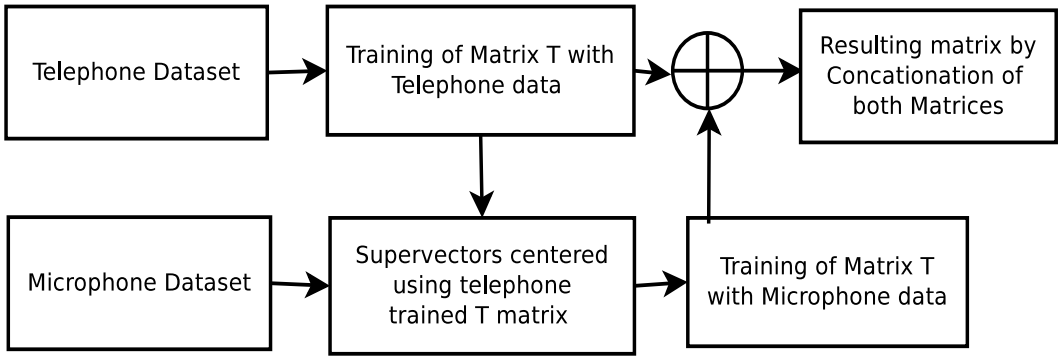

Figure 3.4. Block diagram of microphone suitable I-vector estimation.

projected I-vector using telephone total variability matrix. Using these centered statistics, a new lower rank R_{mic} total variability matrix is trained using only microphone data. It is believed that this lower rank matrix captures residual microphone variability that is not captured by telephone data trained total variability matrix. The model is given in Equation 3.2.

$$\mathbf{M} = \mathbf{m} + \mathbf{T}_{tel}\mathbf{w}_{tel} + \mathbf{T}_{mic}\mathbf{w}_{mic} \tag{3.2}$$

Here, $\mathbf{T}_{tel}$ and $\mathbf{T}_{mic}$ with ranks of R_{tel} and R_{mic} are telephone and microphone trained total variability matrices, respectively. The overall total variability matrix is obtained by concatenation of both $\mathbf{T}_{tel}$ and $\mathbf{T}_{mic}$ matrices.

In this thesis, it is assumed that most of speaker variability is captured by the telephone trained matrix $\mathbf{T}_{tel}$ and the matrix $\mathbf{T}_{mic}$ captures residual microphone channel variabilities. Therefore, it is proposed to use PLDA scoring using R dimensional I-vectors with the following changes. First, all model parameters of PLDA are extracted using telephone data. To capture microphone channel variabilities, the channel variability matrix is further trained with only microphone data and this matrix is concatenated with the telephone data trained one.

3.3. Score Normalization

The last stage of speaker recognition task requires to generate a probabilistic or non-probabilistic score to decide whether the given test utterance is related to the claimed identity or not. It is frequently used in literature that the score distribution is mapped to another one for a couple reasons. First, the decision is made using a predefined threshold whose optimality can depend on each speaker's enrollment. A global optimal speaker independent threshold can be achieved via using one of score normalization technique. For example, use of likelihood ratio in Section 3.2.1, enables the speaker score to be normalized over a score coming from background model. This can be considered as a test utterance level normalization. In this methodology, one can expect that optimal threshold to be unity. An alternative to background model approach is to use a set of impostor models or cohorts [41, 42]. In this method, the score coming from claimed identity model is compared to a set of scores coming from a cohort.

Another reason for score normalization is to compensate channel, environmental and intra speaker variabilities. These variabilities can stem from different environmental conditions, use of different microphones, different transmission of speech data, phonetic content and duration of enrollment. These variabilities can cause mismatch between training and test enrollments, thus causing score mismatch. Score normalization aims to minimize these variabilities at score level. Most commonly used successful normalization techniques include a set of impostor models or test data to map score distribution. These will be discussed in the following sections.

3.3.1. Zero Normalization

Zero Normalization (Z-Norm) is one of the first normalization techniques which is derived from the study in [43]. The purpose is to normalize the score distribution of each speaker to zero mean and unit variance. Enough speaker enrollment data is hard to collect for estimation of normalization parameters, mean and variance. Instead, parameter estimation can be performed over impostor scores. The normalization is

given in Equation 3.3.

$$\hat{S}_s(X^t) = \frac{S_s(X^t) - \mu_s}{\sigma_s} \tag{3.3}$$

Here, $S_s(X^t)$ is raw score for speaker s with the test data of X^t. Impostor utterances are tested under given speaker model s. These test scores are used to estimate in μ_s and σ_s parameters for speaker s. The advantage of this normalization is that normalization is performed off-line at training phase.

3.3.2. Test Normalization

Test Normalization (T-Norm) is proposed in [44] to avoid the acoustic mismatch between test and train data. Z-norm is not capable of preventing this mismatch because the training set is used to train the parameters of normalization. In T-norm, a set of different speaker models is used to generate scores from the test utterance. These scores are used to estimate the normalization parameters. In this sense, this normalization is similar to the cohort approach where a set of different speaker models is used to generate the score. The normalization is given in Equation 3.4.

$$\hat{S}_s(X^t) = \frac{S_s(X^t) - \mu_t}{\sigma_t} \tag{3.4}$$

Here, the parameters, μ_t and σ_t are test enrollment dependent. The disadvantage of this normalization is that estimation of the parameters have to be done at on-line test level.

3.3.3. Symmetric Normalization

Symmetric Normalization (S-Norm) is offered in [36] to deal with score normalization of I-vector approach. In Equation 2.46, denominator of the likelihood can be used to test a collection of utterances and this can be extended to a speaker identification problem [45]. This process can be thought as a normalization and therefore, it

can be expected that t-norm and z-norm are not effective for the PLDA score normalization [36]. ZT-norm, discussed in Section 3.3.4, can disrupt the symmetry between test and training I-vectors because there is no enrollment in I-vector approach as in JFA, GMM/SVM and GMM/UBM methodologies. The s-norm is defined as,

$$\hat{S}(\mathbf{w}_1, \mathbf{w}_2) = \frac{S(\mathbf{w}_1, \mathbf{w}_2) - \mu_1}{\sigma_1} + \frac{S(\mathbf{w}_1, \mathbf{w}_2) - \mu_2}{\sigma_2} \tag{3.5}$$

Here, $S(\mathbf{w}_1, \mathbf{w}_2)$ is the score that is generated between two I-vectors $\mathbf{w}_1$ and $\mathbf{w}_2$. The mean μ_1 and covariance σ_1 parameters are estimated using scores that are generated between $\mathbf{w}_1$ and imposter cohort. Similarly, μ_2 and σ_2 are estimated using scores that are generated between $\mathbf{w}_2$ and the same imposter cohort.

3.3.4. Other Normalizations

Apart from t-norm and z-norm, variants of these normalization techniques have been proposed. ZT-Norm and TZ-Norm are combinations of both t-norm and z-norm. T-norm followed by z-norm is referred as TZ-Norm and t-norm followed by z-norm is referred as ZT-norm [46]. In [47, 48], it is shown that ZT-norm is an effective method in scoring in JFA approach but this is not the case for TZ-norm. H-norm is offered by [49] to normalize difference in handset like carbon and electrolet. It is a similar approach with z-norm but the difference is that instead of speaker dependent normalization, handset dependent normalization parameters are used. In the HT-norm, handset normalization parameters are estimated by testing each test data with set of impostor handset models.

3.4. Evaluation Metric

Speaker recognition task requires a binary decision, either accept or reject, using the generated output score. This induces two different errors, false alarm and miss. Miss occurs when a valid identity is not accepted and false alarm occurs when an invalid identity is accepted. Probability of miss is obtained by the ratio of number of

falsely rejected speaker tests to total number of correct trials. Similarly, probability of false alarm is obtained by the ratio of number of falsely accepted speaker tests to total number of impostor trials. These two types of error depend on the selected threshold value. When threshold is low, system will have low miss rate and high false alarm rate. Similarly, when the threshold is high, system will have high miss and low false alarm rate.

The relative trade-off between false alarm and miss or correct detection rates can be depicted in a single plot which is referred to as Detection Error Trade-off (DET) curve. Each point in the curve represents system performance for the given distinct decision threshold. In the literature, Receiver Operating Curve (ROC) [50] is used to show trade-off between correct detection rate to false alarm rate. It was found more useful to use DET curve in speech applications [51]. In Figure 3.5, a typical DET curve is shown.

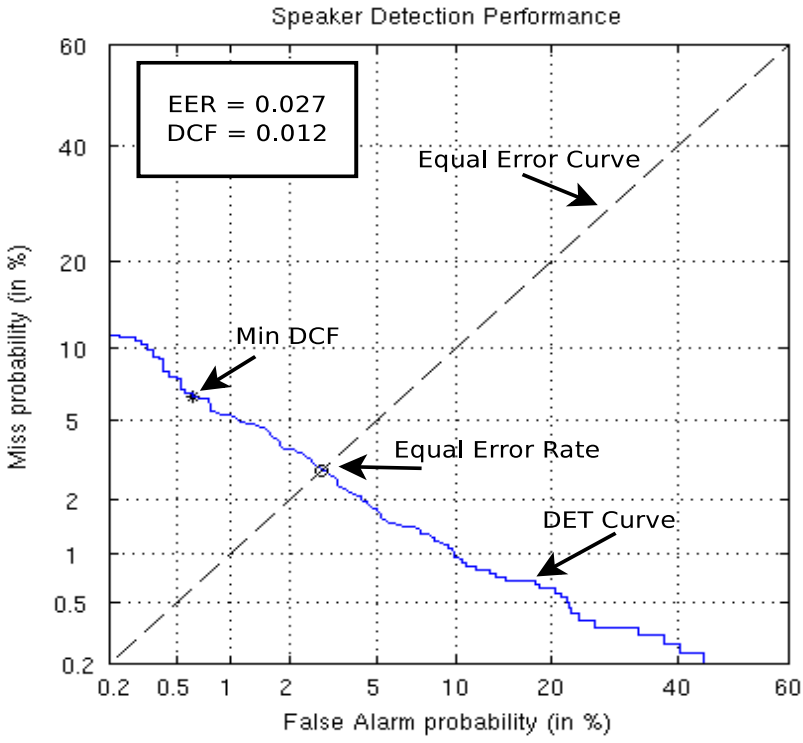

Figure 3.5. An example of a DET curve.

The scales used in the DET curve are logarithmic for convenience. The blue curve

in Figure 3.5 is the trade-off curve for the given system.

When comparing speaker recognition systems, it is difficult to use miss and false alarm rates because these are dependent on each other via the selected threshold. For this reason, threshold independent evaluation metrics, Decision Cost Function (DCF) and Equal Error Rate (EER) are used to determine performance of systems. EER is the value of false alarm and miss rate when threshold is selected to have both of them to be equal as shown in Figure 3.5. Threshold chosen at EER is optimal for the case when number of imposter and correct trials are equal. In NIST SRE, impostor trials are much more than correct trials. To have optimal performance in NIST SRE task, DCF is used as defined in Equation 3.6 [52].

$$C_{Det} = C_{Miss}P_{Miss}P_{Target} + C_{FalseAlarm}P_{FalseAlarm}(1 - P_{Target}) \tag{3.6}$$

P_{Target} is the prior likelihood of an utterance spoken by the correct speaker. C_{Miss} and $C_{FalseAlarm}$ are cost incurred by miss and false alarm, respectively. P_{Miss} and $P_{FalseAlarm}$ are probability of miss and false alarm rate at chosen threshold level. The cost and prior parameters used in NIST SRE 2010 for the specified test conditions are given in Table 3.1. The parameters used in other conditions are used in previous NIST SRE, so it will be referred as oldDCF.

Table 3.1. Cost and prior parameters used in NIST SRE 2010.

	C_{Miss}	$C_{FalseAlarm}$	P_{Target}
core and 8conv/core test	1	1	0.001
Other conditions (oldDCF)	10	1	0.01

In order to improve the meaning of the cost, it is divided by the minimum cost that obtained without processing output scores. This is given in Equation 3.7 [52].

$$\begin{aligned} C_{Default} &= \min\left(C_{Miss}P_{Target}, C_{FalseAlarm}(1 - P_{Target})\right) \\ C_{Norm} &= C_{Det}/C_{Default} \end{aligned} \tag{3.7}$$

The optimal cost function is obtained with the threshold satisfying minimum DCF value and referred to as minDCF.

3.5. Speaker Verification Experiments

In the first part of the experiments, the NIST 2010 SRE submission of Boğaziçi University for the first time is presented. The submission included two different baseline systems for the core test (core-core condition). The first system uses a GMM/UBM methodology whereas the second one uses the SVM-based GMM supervector approach. The following sections describe these two systems. In the second part of the experiments, I-vector analysis with PLDA scoring is studied for the NIST SRE 2010 core conditions. The system is based on the work done in [53]. A new approach in use of PLDA with sparse microphone data is presented.

3.5.1. Baseline Systems for NIST 2010 SRE

The NIST 2010 SRE submission of Boğaziçi University includes two different systems for the core test (core-core condition). The first system uses a GMM/UBM methodology whereas the second one uses the SVM-based GMM supervector approach. The following sections briefly explain these two systems and the computational resources used.

3.5.1.1. GMM/UBM Baseline System. MM/UBM Baseline System is implemented by our group member Erinç Dikici. GMM/UBM implementation is based on the BioSecure Reference System BECARS/HTK [54], which utilizes HTK [55] for feature extraction, UNIANAL [56] for pitch, energy determination and voice activity detection and BECARS [57] for GMM modeling and scoring. 34 dimensional cepstral coefficients (16 MFCC + 16 Δ + energy + Δ-energy) are extracted out of every 20 ms of speech with 10 ms overlaps. Voice activity detection is applied by bi-Gaussian modeling of the energy component. The cepstral vectors are normalized and frames corresponding to silence are deleted. GMMs with 1024 components, having diagonal covariance, is

trained. UBM is constructed using 27 hours of speech from NIST SRE 2008 data. MAP adaptation is used to obtain speaker models. The relevance factor is chosen as 14. Only the means of the components are adapted; covariance and weights are copied from the UBM. Log-likelihood ratio scores are calculated for each test utterance. Final decisions on whether a test utterance is accepted or rejected are given by thresholding with an appropriate value which minimizes DCF over the NIST SRE 2008 test data. Intel® Xeon® CPU E5450 @ 3.00GHz processor including 56 cores in total is utilized during computations. Feature extraction runtime, training and total test times are 40h, 960h and 4888h, respectively.

3.5.1.2. GMM/SVM Baseline System. The system is an SVM baseline setup which uses GMM supervector features. GMMs with 2048 components are trained for both male and female cases separately. Supervectors constructed using the stacked mean vectors of these components are used in the support vector machine as input features. SPro [58] for feature extraction, FIR Echo Canceller [59] for echo cancellation and SVMTorch [60] for SVM training and testing are utilized in this setup. 34 dimensional features are extracted, having 16 MFCC, 16 Δ, energy and the Δ-energy. The lower and higher cutoff frequencies are 300 and 3140 Hz, respectively. 20 ms Hamming window is used with 10 ms increments. ASR files are utilized to remove non-speech segments. Features are mean subtracted and normalized using a three second window. A 2048 mixture GMM is trained on about 12-14 hours of speech data for male and female universal background models using SRE 2008 data. Only half of the short2 and short3, core train and test files, are utilized in UBM training. The other half is used for SVMs negative examples training. 1604 male negative examples and 2217 female negative examples are adapted from the UBM. These negative examples are used to train impostor speaker models. All models are adapted from the UBM using MAP adaptation. Relevance factor is taken as 16. Using these adapted models, speaker SVM models are trained using the supervector linear kernel. Negative examples are obtained through half of the SRE08 short2 and short3 data. In the SVM training we have 1 positive example for the speaker against 1604 negative examples for male and 2217 negative examples for female cases. SVM decision output values are calculated

as final scores. The threshold value for t/f decisions is selected via an analysis over the SRE04 data. Intel® Xeon® CPU E7320 @ 2.13GHz processor including 32 cores in total is utilized during computations. UBMs are trained with 12h and 12.4hours of speech data and computation took 173.6h and 180.1h for male and female cases, respectively. 458.04 hours of impostor training data is used for both male and female cases separately. Total training time is 664.3 hours. All core test evaluation is processed in 1623.52 hours.

3.5.1.3. GMM/UBM and SVM Fusion. The third system is a basic fusion strategy between the outputs of GMM/UBM and SVM systems. For both setups the score threshold is set to 0 and fusion scores are calculated by averaging the systems scores. The hard decision returns true if both systems accept the utterance, if not, it returns false.

3.5.1.4. Results. The DET curves for condition 1 and 5 in NIST SRE 2010 is presented in Figure 3.6 and 3.7. The other seven conditions are presented in Appendix B. However, there is no special attempt for cross channel and different vocal effort conditions. Only, equal balance between interview and telephone data is considered.

It is apparent from minDCF and the DET curves that discriminative SVM/GMM approach performs better GMM/UBM approach. The performance of fusion system is in between the two systems. This is expected because the fusion scheme is not established based on performance of both systems on a common data set. Importance is given primarily on establishing systems that have minimum DCF and calibration of systems are well achieved. Therefore, incorrect selection of the threshold caused the actual DCF to be much higher than the minimum achievable DCF.

Satisfactory performance with GMM/UBM and GMM/SVM baseline systems for a first-time submission is achieved. The weakness of the systems is that it lacks any channel compensation and score normalization.

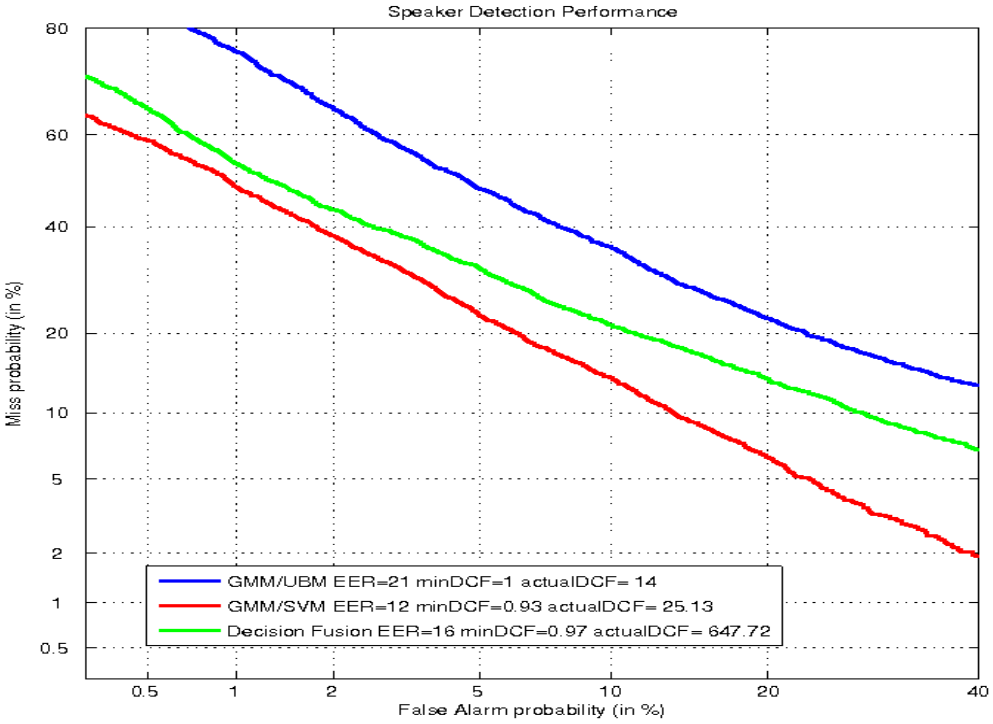

Figure 3.6. DET curve for core condition 1.

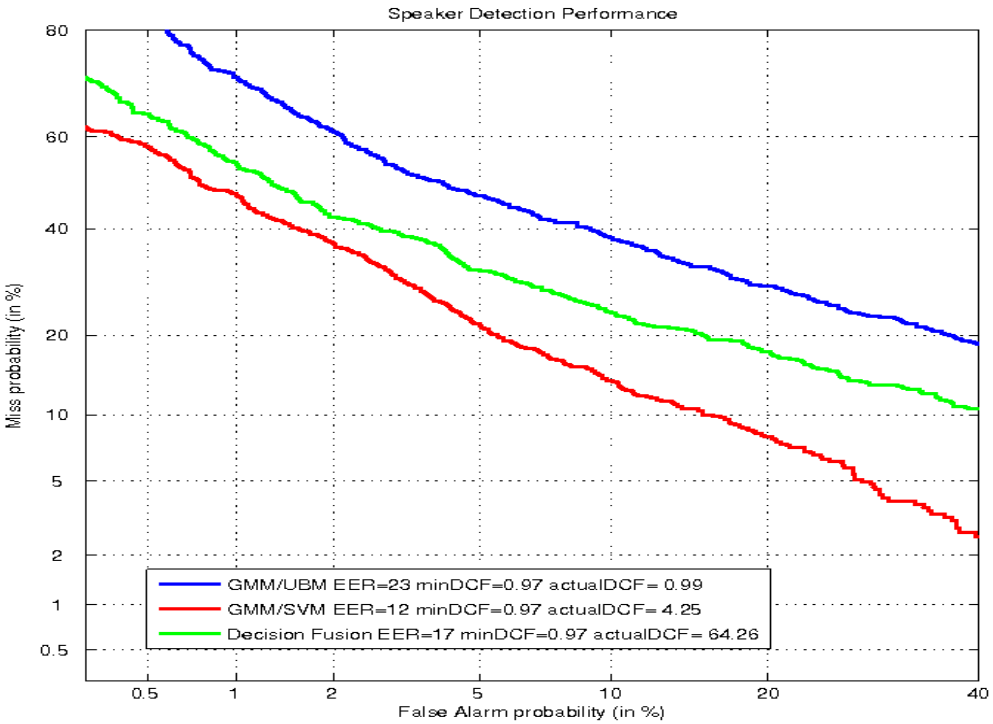

Figure 3.7. DET curve for core condition 5.

3.5.2. I-vector Analysis with PLDA Scoring for Microphone Data

This part of the experiment is continuation of the work in [53]. The aim in this part of the experiments is to determine the best I-vector scoring methodology in microphone core conditions of NIST SRE 2010. Two types of I-vector extractors are trained: standard total variability matrix training for telephone data and I-Vector extractor for microphone data. I-Vector extractor for microphone data is extracted in the way described in Section 3.2.3. Robust estimated total variability matrix $\mathbf{T}_{tel}$ has a rank of R_{tel} and total variability matrix $\mathbf{T}_{mic}$, that is estimated with centralized statistics, has a rank of R_{mic}. Comparison of the performance of I-Vector extractor for microphone data and standard I-vector extractor that is trained with only telephone data is unfair because the former has a rank of $R_{tel} + R_{mic}$, on the other hand latter has a rank of R_{tel}. The performance gain, if any, may stem from the higher dimensionality of the matrix, thus putting the former one in an advantageous position. In order to have a fair comparison, system performance of the concatenated $\mathbf{T}$ matrix which is trained only with telephone data is also reported. All three types of I-vector extractors and their ranks are given in the Table 3.2.

Table 3.2. Training schemes for I-vector extractor.

$[\mathbf{T}_{tel}\ \mathbf{T}_{mic}]$	Tel400	Tel400+Mic200	Tel400_Tel200
$\mathbf{T}_{tel}$ (rank - data)	400 - Tel	400 - Tel	400 - Tel
$\mathbf{T}_{mic}$ (rank - data)		200 - Mic	200 - Tel

To clarify any ambiguity, microphone data refers to both telephone and interview data which are submitted through use of microphone and these are abbreviated as "Mic" and "Int", respectively. Three types of scoring methods are considered, WCCN / LDA with cosine distance, PLDA and PLDA suitable for microphone data. The rank of all matrices in these methods are considered to be half of the dimension of I-vectors, i.e 200 for rank 400 $\mathbf{T}_{tel}$ matrix, except that $\mathbf{W}$ is full rank by definition. Types of data used in the training of parameters of the systems WCCN, LDA and PLDA are studied to determine what combination of data usage has the best performance

gain. Although results of all types of experiment are reported in the detailed results in Appendix A, some of them have no meaning, i.e. PLDA training with only microphone data in telephone only conditions. In the PLDA suitable for microphone data case, first classical PLDA is trained. Then, channel variability matrix is continued to be trained with only area specific data. This newly trained channel variability matrix is concatenated to the one classically trained. For example, in "tel+ mic+int:tel+ mic+int" case first classical PLDA is trained using only telephone, microphone and interview data. Additional three channel matrices are further trained using telephone, "mic" and "int" data in separately. Finally, three more matrices are concatenated to the channel variability matrix.

All types of I-vector scoring methods and the data used in training is given in Table 3.3. The best score normalization is obtained with S-norm in WCCN/LDA case and the results of other normalizations are not reported. However, it is observed that no types of score normalization improved the system performance in the PLDA case.

Table 3.3. I-vector scoring methods with various training data conditions.

WCCN/LDA	PLDA	PLDA + Extra Channels
Tel	Tel	Tel:Tel
Tel - SNorm	Tel + Int+Mic	Tel:Tel + Int+Mic
Tel + Int+Mic	Int+Mic	Tel:Int+Mic
Tel + Int+Mic - SNorm		Tel + Int+Mic:Tel + Int+Mic
Int+Mic		Tel + Int+Mic:Tel
Int+Mic - SNorm		Tel + Int+Mic:Int+Mic
		Int+Mic:Int+Mic

Five core conditions, four of which are associated with microphone data, are considered in the results. The test related to only telephone trials is only performed to check validity of the telephone data only trained matrix $\mathbf{T}$. The types of data used in training and test phases of these five conditions is given in the Table 3.4.

Table 3.4. Types of data in all conditions.

	Condition 1	Condition 2	Condition 3	Condition 4	Condition 5
Train	Int	Int	Int	Int	Tel
Test	Int	Int(different Mic)	Tel	Mic	Tel

The results of all systems are given in following Figure 3.8 to 3.12. Results are given as DCF values whose parameters are given in the Table 3.1. Both new and old DCFs are evaluated.

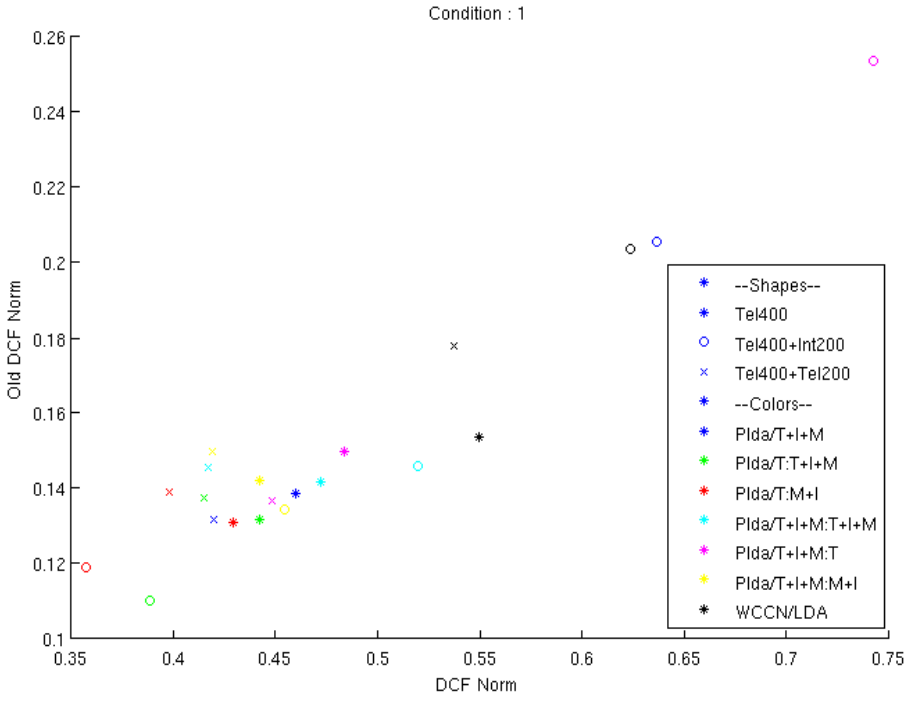

Figure 3.8. Constellation plot for all results in condition 1.

It can be deduced from the experimental results that best results are obtained with PLDA scoring using concatenated channel variability matrices. Concatenation scheme for the best cases is as follows. First, classical PLDA is trained with only telephone data. Then, two best cases are obtained in training further channel matrices with either use of telephone and microphone data or only microphone data.

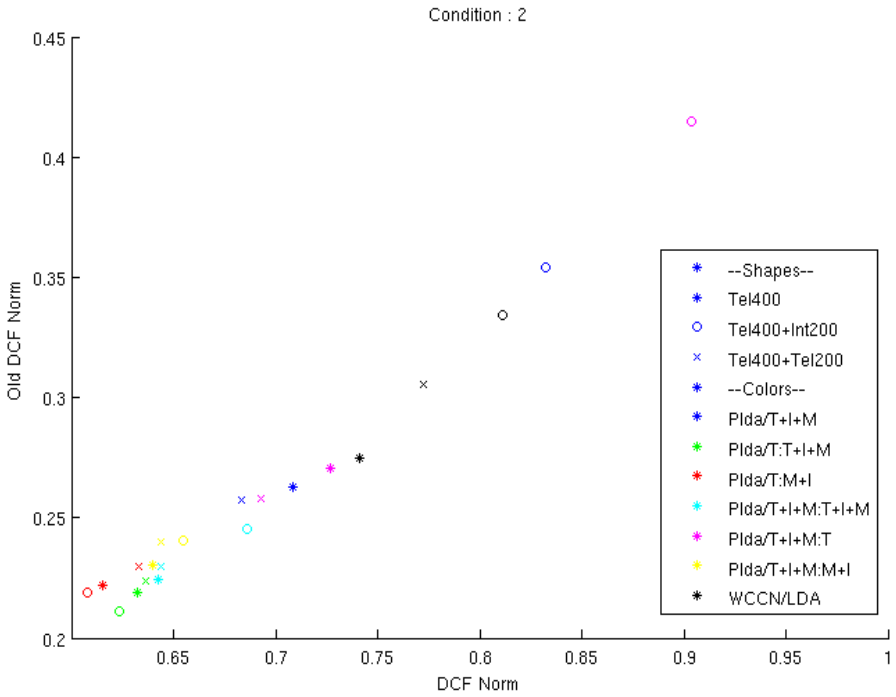

Figure 3.9. Constellation plot for all results in condition 2.

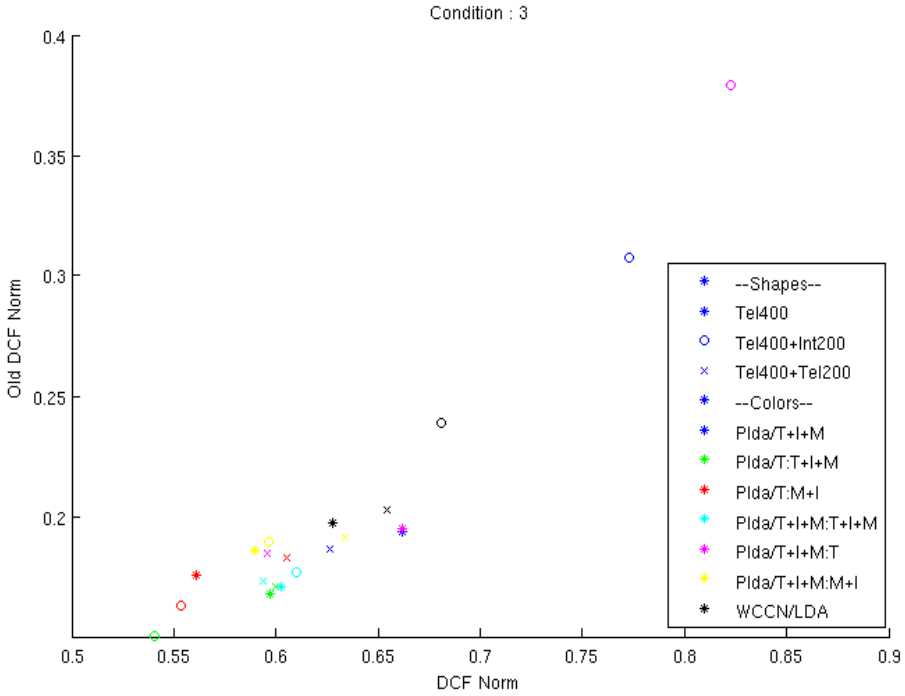

Figure 3.10. Constellation plot for all results in condition 3.

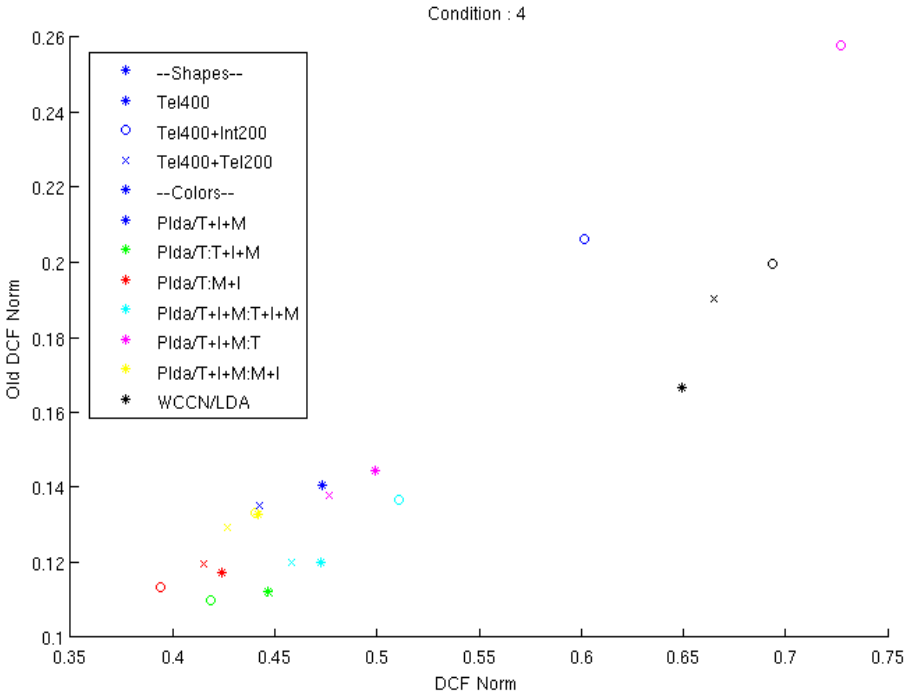

Figure 3.11. Constellation plot for all results in condition 4.

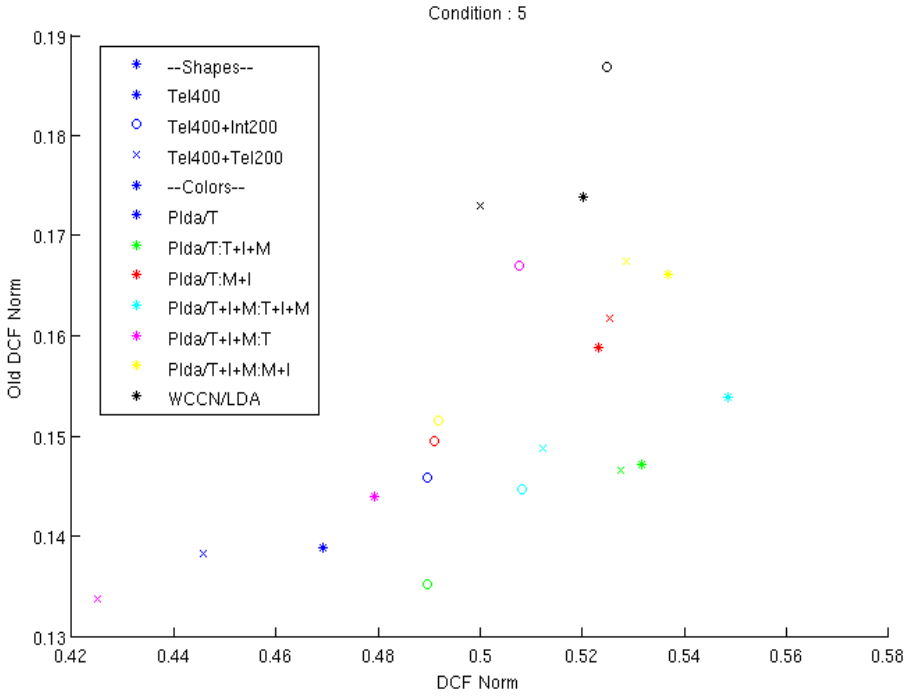

Figure 3.12. Constellation plot for all results in condition 5.

Another improvement is observed in the core condition 5 although it was not intended. Best result is obtained with I-vector extractor Tel400_Tel200. The PLDA training scheme is "Tel+Mic+Int:Tel".

It can be said that most of the speaker variability is captured by Tel400 total variability matrix and remaining speaker and most of the channel variability of "Mic+Int" data is captured by the total variability matrix $\mathbf{T}'$ trained only with microphone data. For this reason, training speaker variability matrix on only telephone data and adding further channel variability matrices trained on only microphone data is the best fit for PLDA scoring of core conditions 1-4.

4. AUDIO DIARIZATION

Audio or speech diarization can be summarized as the "Who spoke when" problem. Detection of speech and non-speech can be considered as a basic segmentation system [61]. Comprehensive segmentation systems can include gender classification and organization of the input source and detection of speech, speaker turn points and narrow band speech. In this chapter, two types of segmentation systems are considered: agglomerative based baseline system and variational Bayesian diarization [62]. In addition, Maximum Likelihood Speech Source Separation is proposed for detection of simultaneous speech segments and identities of speakers in this segment. Lastly, speaker adaptation for Large Vocabulary Continuous Speech Recognition (LVCSR) using the speaker segments via automatic speaker segmentation is discussed. This part of thesis is the continuation of the work done by our group for Turkish broadcast news transcription [63].

4.1. Baseline Audio Diarization Setup

In this work, a standard baseline speaker segmentation system is used. It uses speech segmentation, speaker change detection using BIC, speaker clustering and final Viterbi refinement of speaker turn points. These steps are described in the following subsections.

4.1.1. Speech Detection

The purpose of this step is to find the speech segments in a given audio. This step cannot be implemented using energy based speech segmentation because broadcast audio contains sources like music which can be classified as speech due to high energy content. Instead of energy based speech detection, maximum likelihood classification using GMMs as state output distributions are used. This is a one pass Viterbi segmentation which is illustrated in Figure 4.1.

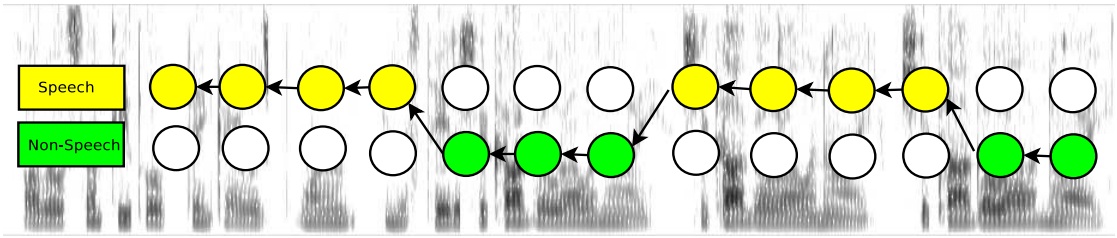

Figure 4.1. Illustration of HMM based speech detection.

There are two states in HMM as in [64] which are speech and non-speech hidden states. The parameters of the HMM are trained using labeled data. Speech state can be further divided into sub-states which are gender, bandwidth, noisy and clean speech states. However, this division approach provided no performance gain in our experiments. Therefore, only a single speech state is train with various types of speech data. The non-speech state is trained with music and any other non-speech segments.

4.1.2. Speaker Turn Point Detection

The purpose of this step is to find change points stemming from different audio sources such as speaker turn points. The general approach to this problem is to use two adjoint windows and calculation of a metric, like KL-2 divergence [65] or Bayesian Information Criteria (BIC) [66], between these windows. When the metric value is higher than a threshold value which is determined using cross validation, then, a new speaker turn point is declared.

Definition of BIC is given in Equation 4.1 and it is illustrated in Figure 4.2. $Num(M)$, K and L represent model complexity, number of data points and likelihoods, respectively.

$$BIC(M) = \log L(\mathbf{X}, M) - \lambda \frac{1}{2} Num(M) \log(K) \tag{4.1}$$

Data in a window and its distribution are given in Equation 4.2. A multivariate

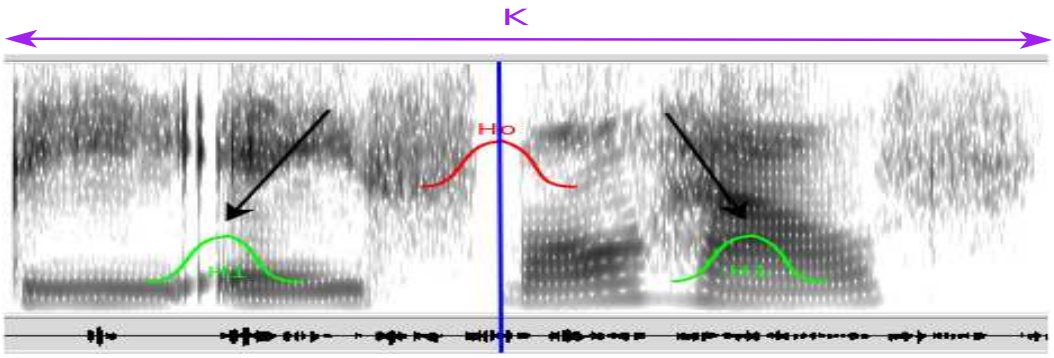

Figure 4.2. Illustration of BIC turn point detection.

Gaussian distribution is trained using the data within the window.

$$H_0 : \mathbf{x}_1 \dots \mathbf{x}_K \sim \mathcal{N}(\mu, \mathbf{\Sigma}) \tag{4.2}$$

Similarly, two Gaussian distributions for the ith turn point using the data in the two different windows are given in Equation 2.16.

$$H_1 : \mathbf{x}_1 \dots \mathbf{x}_i \sim \mathcal{N}(\mu_1, \mathbf{\Sigma}_1); \;\; \mathbf{x}_{i+1} \dots \mathbf{x}_K \sim \mathcal{N}(\mu_2, \mathbf{\Sigma}_2) \tag{4.3}$$

Difference of BIC for both hypothesis is given in Equation 4.4.

$$\Delta BIC = K \log(|\mathbf{\Sigma}|) - K_1 \log(|\mathbf{\Sigma}_1|) - K_2 \log(|\mathbf{\Sigma}_2|) \gtrless \lambda_{threshold} \tag{4.4}$$

Using ΔBIC and an appropriate threshold, $\lambda_{threshold}$, a possible turn point is decided. The Equation 4.4 is simplified version of BIC in order to reduce computation. Simplification is due to use of only center of a window as a possible turn point. Therefore, the ith turn point is fixed to middle of the window. However this may cause to miss possible turn points if classical BIC approach in finding turn point is used here. In order not to miss possible turn points, windows are slided on very small intervals. However the proposed approach requires to set a minimum speech segment constraint to prevent having many speaker turn points around the actual turn point.

4.1.3. Clustering

In this step, segments that are generated by the same sources or speakers are clustered. Commonly, bottom up Hierarchical Agglomerative Clustering (HAC) method is used for this purpose. For each cluster, GMM is adapted from the UBM via MAP adaptation. The UBM is trained using the speech segments from all sources. Then, distances between clusters are calculated. The closest clusters are assumed to be generated by the same speaker and they are merged as illustrated in Figure 4.3.

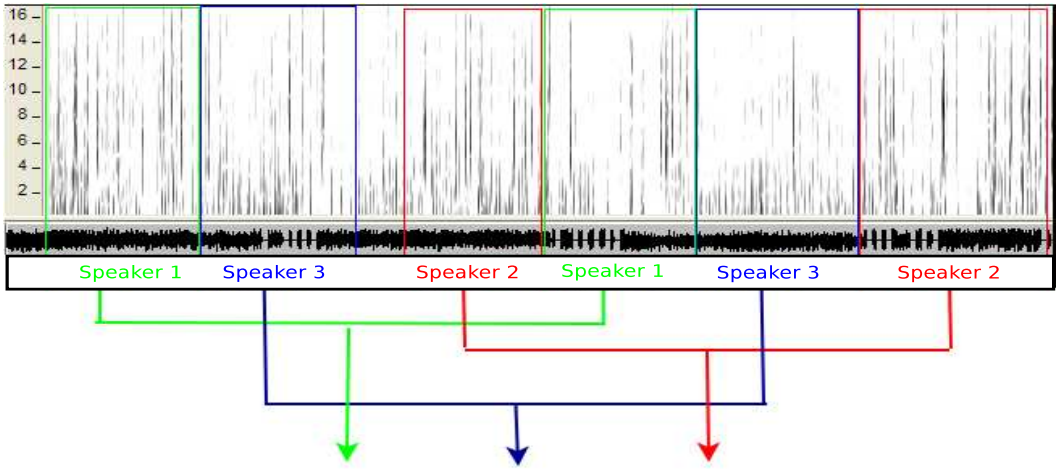

Figure 4.3. Illustration of HAC.

Cross Likelihood Ratio (CLR) [67], given in Equation 4.5, is used as a distance measure between clusters, namely speakers s_1 and s_2.

$$CLR(\mathbf{x}_1, \mathbf{x}_2) = \log \left(\frac{L(\mathbf{x}_1|s_2)L(\mathbf{x}_2|s_1)}{L(\mathbf{x}_1|\text{UBM})L(\mathbf{x}_2|\text{UBM})} \right) \tag{4.5}$$

Here, $\mathbf{x}_1$ and $\mathbf{x}_2$ are the all speech data of speaker s_1 and s_2, respectively. The highest likelihood ratio between two clusters are assumed to be from same speaker. At each iteration, these two clusters' data is used to adapt new GMM from UBM via MAP. Clustering is stopped when CLR is lower than a threshold.

4.1.4. Re-segmentation

Using the clusters and their estimated models, iterative Viterbi re-segmentation is performed to better align speaker turn points. The cluster models are used as hidden states in the HMM and this is illustrated in Figure 4.4.

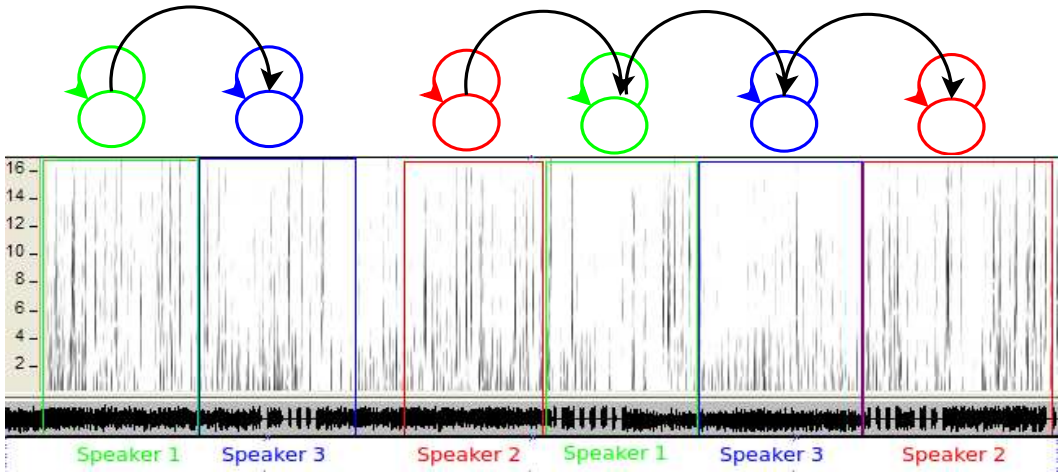

Figure 4.4. Illustration of Viterbi re-segmentation.

This approach is more successful than trusting the speaker turn points that are obtained with BIC approach because the prior speaker models are utilized in this method that was not available in BIC case. This step further reduces error rate substantially. Another successful method at this level is soft speaker clustering [62]. In this method, likelihoods of each speakers are calculated for each segment. Using these likelihoods, the posterior probability of a speaker speaking in a speech segment is calculated. Statistics are extracted from the segments and are weighted according to the posterior probabilities. For each speaker, the weighted statistics are used to adapt the GMM. The advantage of this method that it avoids hard decision as in Viterbi case.

4.2. Overlapping Speaker Segmentation

In the previous section, a standard speaker segmentation system is presented. However, this system is not capable of assigning multiple speakers to speech segments. This assignment is essential because some audio contains speech segments in which

there are more than one speaker. When NIST Rich Transcription evaluation metric is used, it is seen that all the unassigned overlapping speech segments are considered as missed speaker time error. In the following section, a maximum likelihood speech source separation algorithm is proposed to decompose a samples into two samples to represent two different feature vectors that are generated by two different speakers. The decomposition is limited to two because it is rarely the case that three or more speakers speak at the same time. In the second part of the proposed method, it is determined whether there is overlapping speech or not, as well as identities of speakers using the decomposed samples.

4.2.1. Maximum Likelihood Speech Source Separation

After an initial segmentation is performed, segments are assigned to single speakers. For example, some segments are assigned to speaker s_i in a given audio. After this point, it should be checked whether speakers s_j $\forall j \neq i$ also spoke at the segments of speaker s_i. For this purpose, maximum likelihood speech source separation is proposed. In this method, the acoustic features in each segment of speakers s_i, $i = 1 \dots N$ are decomposed for speakers s_i and s_j, $\forall i \neq j$ and $j = 1 \dots N$ where N is the total number of speakers after the initial segmentation is performed.

Feature decomposition via maximum likelihood is as follows. Consider the observation acoustic feature vector o. For speakers s_1 and s_2, the following decomposition model is assumed.

$$o = x_1 + x_2 \tag{4.6}$$

The clusters are modeled with UBM adapted GMMs. The objective function is obtained as the likelihood of decomposed vectors x_1 and x_2 and the constraint in Equation 4.6. Direct use of the likelihood function causes maximum likelihood solution to has an intractable form. Therefore, a lower bound approximation using Jensen's inequality

is used. The approximated objective function is given in Equation 4.7.

$$L^* = \sum_{c=1}^{C} p(\omega_{1c}) \frac{-(x_1 - \mu_{1c})^2}{2\sigma_{1c}^2} + \sum_{c=1}^{C} p(\omega_{2c}) \frac{-(x_2 - \mu_{2c})^2}{2\sigma_{2c}^2} + \lambda(x_1 + x_2 - o) \tag{4.7}$$

The resulting estimation is given in Equation 4.9 using the conventions in the Equation 4.8. The equations are of the single dimensional form, but one can easily switch it to multivariate case because covariance is in diagonal form.

$$\mu_i' = \sum_{c=1}^{C} p(\omega_{ic}) \frac{\mu_{ic}}{\sigma_{ic}^2} \qquad \sigma_i' = \sum_{c=1}^{C} p(\omega_{ic}) \frac{1}{\sigma_{ic}^2} \tag{4.8}$$

$$x_1 = \frac{\lambda + \mu_1'}{\sigma_1'} \qquad x_2 = \frac{\lambda + \mu_2'}{\sigma_2'} \tag{4.9}$$

The solution can be further converted into following form.

$$x_1 = \frac{\sigma_2' o - \mu_2' + \mu_1'}{\sigma_1' + \sigma_2'} \qquad x_2 = \frac{\sigma_1' o - \mu_1' + \mu_2'}{\sigma_1' + \sigma_2'} \tag{4.10}$$

4.2.2. Overlap Detection

The purpose of this part is to determine whether the observation is generated by one or two different speakers using the proposed decomposition model in Equation 4.10. Following logarithmic likelihood function is considered for this purpose.

$$L(\mathbf{x}_1, \mathbf{x}_2) = \log \left(\frac{L(\mathbf{x}_1|s_1) L(\mathbf{x}_2|s_2)}{L(\mathbf{x}_1|\text{UBM}) L(\mathbf{x}_2|\text{UBM})} \right) \tag{4.11}$$

When this likelihood exceeds a predefined threshold, labels of the two speakers are assigned to the segment. In order to test the performance of vector decomposition algorithm presented in the previous section, likelihood in Equation 4.11 is also tested with only observation vector, o instead of decomposition vectors x_1 and x_2.

4.3. Factor Analysis Based Audio Diarization

Factor analysis based audio diarization that is studied in this section is adapted from the work in [62]. Apart from variational Bayesian approach, an I-vector analysis approach to this problem is studied. First, an initial diarization via the baseline system presented as in previous section or a random initialization is assumed. Speakers are represent as in the JFA model or I-vector analysis in Equation 2.29. JFA model can be obtained as in Equation 2.29 via concatenation of both lower rank channel and speaker factor loading matrices if both matrices are of the same rank and if residual speaker variability matrix $\mathbf{D}$ is not used in the JFA model. Instead of hard assignment to speech segments, soft assignment is made on segments using posterior probabilities. Using the soft assignments, speaker models are trained or extracted using synthesized Baum-Welch statistics for JFA and I-vector analysis cases, respectively. These statistics are synthesized for each speaker using segment posterior probabilities and Baum-Welch statistics extracted from the segments.

In [68], $\log \hat{q}_{ms}$ is defined as in Equation 4.12.

$$\log \hat{q}_{ms} = \log \pi_s p(m|\mathbf{w}_s) - \frac{1}{2}\mathrm{tr}(\mathbf{T}^T\mathbf{N}^m\mathbf{\Sigma}^{-1}\mathbf{T}\mathbf{L}_s^{-1}) \tag{4.12}$$

Here, m represents a speech segment. $\mathbf{N}^m$ is diagonal matrix whose diagonals $N_c\mathbf{I}$ $(c = 1, \ldots, C)$. It is extracted from the segment m. Similarly, $\overline{\mathbf{F}}^m$ will be referred as first order statistic supervector that is extracted from the segment m. The latent variable $\mathbf{w}$ in Equation 2.29 is assumed to be normally distributed with mean $\mathbf{w}_s$ and variance $\mathbf{L}_s^{-1}$ for the speaker s. $\mathbf{w}_s$, which is also referred as I-vector, is the point estimate of $\mathbf{w}$. π_s is the prior segment probability which is initially obtained as one over total number of speakers found in the segmentation. Conditional segment (m) probability is given in Equation 4.13.

$$\begin{aligned} \log p(m|\mathbf{w}_s) &= \sum_{c=1}^{C} N_c^m \log \frac{1}{(2\pi)^{D/2}|\mathbf{\Sigma}_c|^{1/2}} - \frac{1}{2}\mathrm{tr}(\mathbf{\Sigma}^{-1}\overline{\mathbf{S}}^m) \\ &+ \mathbf{w}_s^T\mathbf{T}^T\mathbf{\Sigma}^{-1}\overline{\mathbf{F}}^m - \frac{1}{2}\mathbf{w}_s^T\mathbf{T}^T\mathbf{N}^m\mathbf{\Sigma}^{-1}\mathbf{T}\mathbf{w}_s \end{aligned} \tag{4.13}$$

$\overline{\mathbf{S}}$ is diagonal matrix whose diagonals are $\overline{\mathbf{S}}_c$ $(c = 1, \ldots, C)$. Then the update formula for the posterior probability of speaker s and segment m is given in Equation 4.14.

$$q_{ms} = \frac{\hat{q}_{ms}}{\sum_{s=1}^{S} \hat{q}_{ms}} \tag{4.14}$$

After the posterior probabilities are calculated for each speaker Baum-Welch statistics are synthesized as follows.

$$\mathbf{N}(s) = \sum_{m=1}^{M} q_{ms} \mathbf{N}^m \qquad \overline{\mathbf{F}}(s) = \sum_{m=1}^{M} q_{ms} \overline{\mathbf{F}}^m \tag{4.15}$$

Then, the update formula for $\mathbf{w}_s$ and $\mathbf{L}_s$ is given in Equation 4.16.

$$\mathbf{L}_s = \mathbf{I} + \mathbf{T}^T \mathbf{\Sigma}^{-1} \mathbf{N}(s) \mathbf{T}; \quad \mathbf{w}_s = \mathbf{L}_s^{-1} \mathbf{T}^T \mathbf{\Sigma}^{-1} \overline{\mathbf{F}}(s); \quad \pi_s = \frac{1}{M} \sum_{m=1}^{M} q_{ms} \tag{4.16}$$

The overall likelihood is given in Equation 4.17 where R is the rank of the space.

$$L = \frac{1}{2} \left(RS - \sum_{s=1}^{S} \left(\log |\mathbf{L}_s| + \operatorname{tr} \left(\mathbf{L}_s^{-1} + \mathbf{w}_s \mathbf{w}_s^T \right) \right) \right) + \sum_{m=1}^{M} \sum_{s=1}^{S} q_{ms} \log \hat{q}_{ms} - q_{ms} \log q_{ms} \tag{4.17}$$

When overall likelihood is converged, hard assignments speakers to speech segments can be applied instead of using posterior probabilities as in the Equation 4.14. Also, the overall likelihood can be used in the process of determination of overall number of speakers in the audio source. For example, speaker clusters can be merged until there is no increase in the the overall likelihood.

A simplification to this variational Bayesian approach can be made by directly using the point estimate of the hidden variables and generate a score as in Section 2.2.4.1. Again, posterior probabilities can be calculated as in Equation 4.14.

4.4. Speaker Adaptation for LVCSR via MLLR

Statistical speech recognition systems consist of acoustic and language models. Acoustic models are based on HMM whose states are GMMs. Training HMM parameters requires large amounts of data. It is easier to gather sufficient data in training of speaker independent models but they perform worse than speaker dependent ones [16]. One possible solution to this problem is adaptation of speaker independent parameters of HMM to each speaker whose adaptation data can be obtained through automatic segmentation of speech. MLLR method is used to estimate adaptation matrix and to obtain speaker adapted HMM for the task of LVCSR of Turkish Broadcast News. Adaptation is performed on only mean vector of GMM because it is believed that difference between speaker are due to mean vector only [16]. The details of estimation is detailed in Section 2.2.1.3.

4.5. Error Metric

There are two types of error after speech diarization is performed. They are missed speech error where non-speech is recognized as speech and false alarm speech where speech is labeled as non-speech. This error is fixed after the speech detection is performed. There are three types of error which are associated with speaker assignments. Missed speaker error is total amount of unlabeled speaker segments including overlapping regions. False alarm speaker error is associated with labeled speaker segments that don't exist in the reference. Lastly, the speaker time is sum of all wrong speaker assignments.

The formal evaluation metric is defined by NIST and well in the Rich Transcription (RT) evaluations [69]. The primary evaluation metric is given in Equation 4.18 where m represents a speech segment and D_m is the duration the m segment.

$$\text{ERROR} = \frac{\sum_m D_m(\max(N_{(Ref,m)}, N_{(Sys,m)}) - N_{(Cor,m)})}{\sum_m D_m N_{(Ref,m)}} \tag{4.18}$$

$N_{(Ref,m)}$,$N_{(Sys,m)}$ and $N_{(Cor,m)}$ are number of reference speakers speaking in segment

m, number of system speakers speaking in segment m and number of correctly matched speakers speaking in segment m, respectively.

Formal definition of missed speaker time, false alarm speaker time and speaker time are given in Equation 4.19, 4.20 and 4.21, respectively.

$$\sum_m D_m(N_{(Ref,m)} - N_{(Sys,m)}) \qquad \exists m : N_{(Ref,m)} > N_{(Sys,m)} \tag{4.19}$$

$$\sum_m D_m(N_{(Sys,m)} - N_{(Ref,m)}) \qquad \exists m : N_{(Sys,m)} > N_{(Ref,m)} \tag{4.20}$$

$$\sum_m D_m(\min(N_{(Ref,m)}, N_{(Sys,m)}) - N_{(Cor,m)}) \tag{4.21}$$

Addition of these three separate errors is equivalent to the overall error as in Equation 4.18.

4.6. Experiments

In this section, first the speaker segmentation setups that are common to all experiments are described. Then, experiments on speaker adaptation, overlapping speaker segmentation and factor analysis based segmentation are introduced in the following sections.

4.6.1. Speaker Segmentation Setups

The baseline speaker segmentation system in Section 4.1 is used for speaker adaptation of acoustic models and overlapping speech detection experiments. A comparative study between the baseline and the factor analysis based system is considered.

The setup of the baseline speaker segmentation system is as follows. Broadcast news programs are sampled at 16 kHz frequency. Feature vectors are extracted using 20 ms windows using Hamming window. A sliding window over 10 ms duration is used. Frequencies between 300 and 8 kHz are analyzed. Feature vectors are 34 dimensional consisting of 16 MFCC and energy coefficient and its first derivatives.

Two state HMM with speech and non-speech states, are used in Viterbi decoding of speech to detect speech boundaries. States are modeled with GMM consisting of 128 mixing components using 30 minutes of manually labeled data. Minimum speech duration is assumed to be one second.

In BIC detection of speaker turn points, three seconds of a window frame is used with a 20 ms sliding window. The shortest speech segment is considered to be at least one second in duration. Using one whole broadcast news program, UBM with 64 mixing and 32 mixing components are trained for the task of speaker adaptation and overlapping speaker segmentation, respectively. Speaker recognition requires higher mixing components in the training of GMM. However, it is known that 32-64 mixing components are sufficient in training GMM of different speakers who share similar channel characteristics as in broadcast news [62]. All segments are represented by GMM that are MAP adapted from UBM. Adaptation is performed only on the mean vector. Relevance factor is taken as 14. At the last step, Viterbi decoding is performed in three iterations to refine segment boundaries.

4.6.2. Speaker Adaptation Experiments

In the experiments, six different broadcast news programs, which make three hours of audio data in total, are used. There are five to twenty different speakers in each bulletin. Segmentation is performed for each bulletin and the speaker segments are obtained. Using speaker segmented data, MLLR speaker adaptation is performed in three iterations to obtain speaker dependent acoustic models. Only mean vectors are adapted. Then, LVCSR of broadcast news is performed. LVCSR is also tested with speaker independent acoustic models to compare performance of the developed system. 39 dimensional feature vectors consisting of 13 MFFC and its first and second derivatives are extracted for the adaptation experiments.

4.6.2.1. Experimental Setup. Three different LVCSR systems and four different speaker adaptation systems are used in the experiments resulting in twelve different results.

Difference in LVCSR systems is due to acoustic and language models. 200 hours of speech and 350 million word data, 200 hours of speech and 200 million word data, 350 hours of speech and 350 million word data are used for training of acoustic and language models in experiments 1, 2 and 3, respectively.

Difference in speaker adaptation is due to different speaker segmentation methods in obtaining the speaker's segmented data. The baseline system results are obtained by using speaker independent acoustic models. No segmentation and adaption is applied. In the fully automatic segmentation setup, automatic segmentation system is used to obtained the speaker data. In the other segmentation setup, manually labeled speaker segmented data is utilized. In this way, maximum improvement in LVCSR that can be achieved with speaker adaptation using perfect segmentation setup is observed. Half segmentation (semi automatic) is the last setup. In this setup, speaker turn points are manually labeled but speaker assignments to segments are not made. In other word, detection speaker turn point using BIC is done manually but HAC step is automatically performed.

Each experimental setup is tested under 7 different Real Time Factor (RTF). Error metric is taken as Word Error Rate (WER). Intel® Xeon® CPU E7320 @ 2.13GHz processor is used in the experiments.

4.6.2.2. Results. Results of experiments 1-3 are given the following Figure 4.5, 4.6 and 4.7 and fully automatic segmentation setup results of all experiments is given in Figure 4.8.

In can be deduced from the figures that experimental setup 3, trained with most amount of data, has lowest WER at same RTF levels of other experimental setups. As expected, manually labeled segmentation setup has the lowest WER among other segmentation setups in Figure 4.5. Semi automatic segmentation has better performance than the baseline and automatic segmentation systems. When RTF is below nearly 0.5, automatic segmentation system has higher WER. However, its performance gain

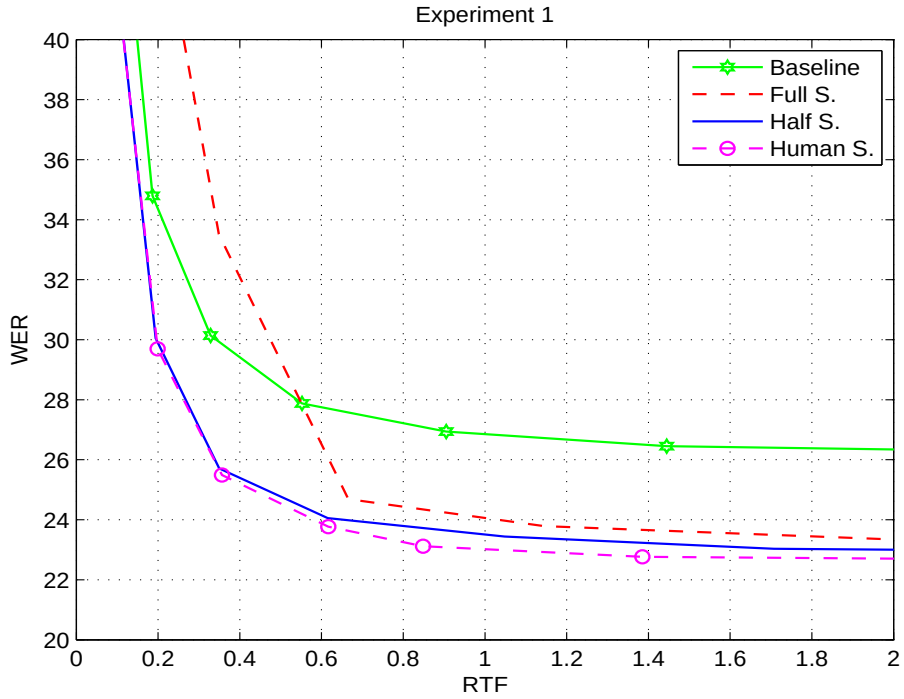

Figure 4.5. Speaker adaptation results for setup 1.

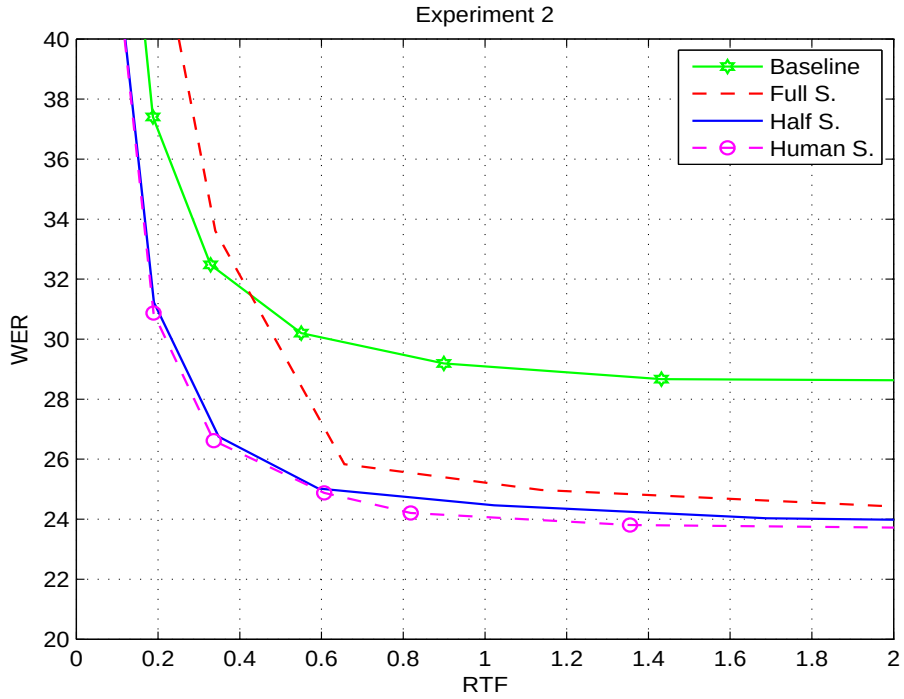

Figure 4.6. Speaker adaptation results for setup 2.

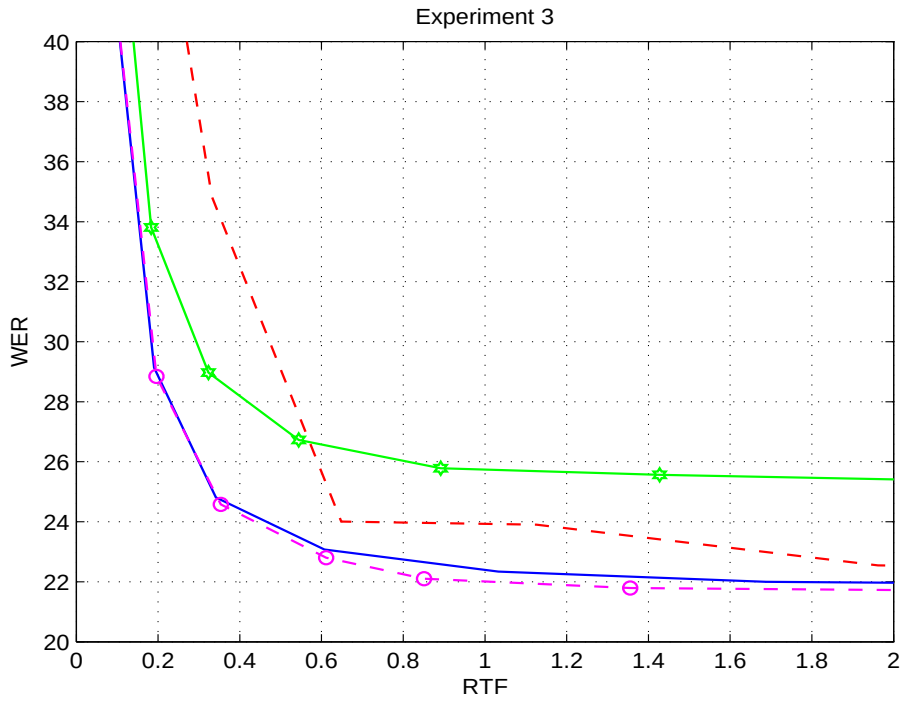

Figure 4.7. Speaker adaptation results for setup 3.

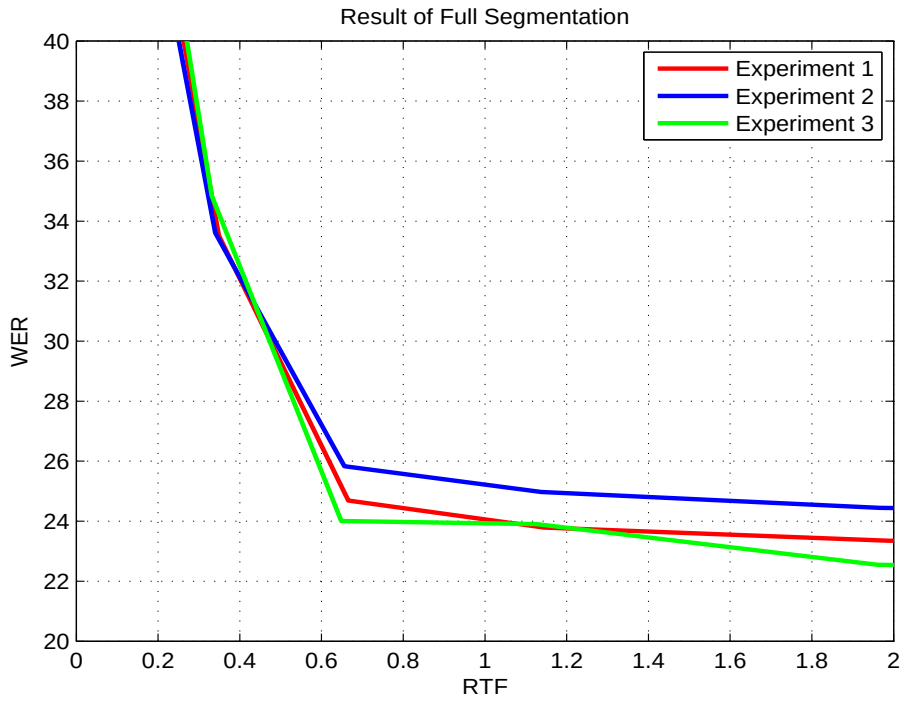

Figure 4.8. Fully automatic segmentation results for all experiments.

is observed at higher levels of RTF. Also, it is observed that automatic segmentation system has higher RTF compared to other case because it requires computationally expensive algorithms like BIC and greedy HAC. Lowest WER is achieved at highest RTF level of experiment 3. At this configuration, WER is 22.4% and 25.4% for automatic segmentation and baseline systems, respectively. Therefore, 3% absolute and 11.8% relative WER reduction is achieved with respect to the baseline system. At the same configuration for experiment 2, WERs are 24.3% and 28.6% for automatic segmentation and baseline systems, respectively. 4.3% absolute and 28.6% relative WER reduction is achieved for this case. It also observed in this configuration that the error reduction gain obtained from manually labeled segmentation setup than automatic segmentation setup is around 0.6% in WER. Maximum attainable gain in WER that can be achieved in LVCSR using speaker diarization is almost succeeded with the baseline automatic speaker segmentation setup.

4.6.3. Experiments with Overlapping Speaker Segmentation

Feature decomposition and detection of overlapping speaker algorithm are required to be applied $\binom{N}{2}$ times given a broadcast news program containing N number of speakers. The current problem is already a challenging task, therefore it can be expected to have high error rates when number of speakers is high. In order to have low amount of computation and simplified problem, the program "Yorum Farkı", which contains only two speakers, is used. Collar size at the speaker turn points is determined to be the default value of 0.25 ms.

4.6.3.1. Results. The details of the news program is given in Table 4.1, the result of speech/non-speech detection is given in Table 4.2 and lastly results of three different experiment is given in Table 4.2. In the experiment 1, no detection of overlapping speech is performed. In the experiment 2, overlapping speech detection is performed using observation vectors without applying feature decomposition algorithm. In the last experiment, overlapping speech detection is performed using the proposed decomposition algorithm.

Table 4.1. Summary of the test instance by time.

	Time (s)	Rate (%)
Total Time	1213.1	100
Total Speech	1162.5	95.8
Time in Overlaps	50.6	4.2
Time* Speakers in Overlaps	101.2	8.4

Table 4.2. Result of speech / non-speech detection.

	Time (s)	Rate (%)
Missed Speech	0.0	0
False Alarm Speech	10.7	0.9

Table 4.3. Results of the three experimental setups.

Speaker Error Rates (%)	Experiment 1	Experiment 2	Experiment 3
Scored Speaker Time	104.5	104.5	104.5
Missed Speaker Time	4.3	3.0	3.2
False Alarm Speaker Time	0.9	2.2	1.6
Speaker Error Time	0.7	0.2	0.3
Overall Error Rates	**5.9**	**5.4**	**5.1**

As a result, it is shown that detection of overlapping speech using feature decomposition reduced the total error 5.9% from to 5.1%. On the other hand, detection of overlapping speech without using feature decomposition can only reduced the error rate to 5.4%.

4.6.4. Experiments with Factor Analysis Based Systems

In this section, three different diarization methodologies using JFA, I-vectors and GMMs are reported. The experiments presented here are based on the work in [62]. However, performance of I-vector system was not analyzed in that study. Therefore, it is aimed to analyze performance of an I-vector system in this work. The JFA methodology that is mentioned in this work differs from the one in [62] in one respect. It uses the total variability matrix, that is used for I-vector extractor, as a substitute for concatenation of speaker and channel matrices as in [62]. Thus, model parameters are same for both systems and the performance difference between I-vector and JFA methods are due to their different score calculation. In the following subsections, modeling of the systems, the systems that are used in this study and finally results of the systems for different conditions are presented.

4.6.4.1. System Modeling. The feature vectors used in all systems are extracted with a 10 ms sliding window of 25 ms duration. 39 dimensional feature vectors are obtained by concatenating 13 dimensional MFCC vector and its first and second derivatives. Feature vectors are locally mean subtracted (CMS) and covariance normalized using a 3 seconds window. Short time gaussianization or feature warping are not performed because of the reasons described in [62].

The data used in the experiments are provided by our group [63]. Only the news data that are collected from the same television channel is used for development and test data. Each news bulletin contains 20 to 30 minutes of speech data in average.

All test and training samples contain only speech segments. Therefore, a speech/non-speech segmentation step is not needed. Also, there are no multiple speaker segments in the samples. Therefore, the overall diarization error obtained in these experiments are only due to wrong speaker assignments.

For the HAC using GMMs systems, 32 and 16 mixture component UBMs are

trained using the whole sample in Section 4.6.4.3 and 4.6.4.4, respectively. The whole sample is not sufficient to train 32 mixture component UBM for the latter case.

The UBM for total variability matrix training is trained using 5% of the whole training data. The training data is randomly selected from each utterance to represent large speaker and acoustic variations. UBM consists of 512 Gaussians. Also, 256 and 1024 mixture component UBMs with different ranks of total variability matrix are used to determine the optimal model complexity. Using a small amount of held-out set, some improvement in the performance of the system is observed over 256 mixture UBM case but no improvement is observed over 1024 mixture UBM case.

The whole development data are used to extract speaker statistics for total variability matrix training. The test data can contain speakers who speak various amounts. For this reason, statistics for development data are extracted from different sizes of speech segments. Around 6000 utterances are used in total variability matrix training. The rank of total variability matrix is set to be 100 which is determined via validation.

Rank of LDA matrix is determined to be half of the rank of the total variability matrix which is 50. The same training data that is used for the total variability matrix is used to train LDA and WCCN matrices. Same speakers in different news bulletin are not labeled with the same speaker label. In order to avoid the negative effect of wrong speaker assignments, the following proposition is offered. The classical LDA aims to maximize the objective function in Equation 4.22. $\mathbf{S_b}$ is the between cluster and $\mathbf{S_w}$ are the within cluster matrices and $\mathbf{w}$ is the I-vector.

$$\mathbf{w}\mathbf{S_b}\mathbf{w} - \lambda(\mathbf{w}\mathbf{S_w}\mathbf{w} - 1) \tag{4.22}$$

However, the between and within cluster matrices are estimated for each news bulletin separately in order not to consider same clusters as different clusters. Therefore, the optimization function for this case is as in Equation 4.23 assuming that $\sum_{i=1}^{N} \mathbf{w}\mathbf{S_w}$

where N is the number of news bulletins.

$$\sum_{i=1}^{N} \mathbf{w}\mathbf{S_b}^{i}\mathbf{w} - \lambda(\mathbf{w}\mathbf{S_w}^{i}\mathbf{w} - 1) \tag{4.23}$$

Therefore, the sum of the between and within cluster matrices that are estimated from each news bulletin is used in estimation of the overall LDA matrix. Similarly, the WCCN matrix is estimated using the sum of the within cluster matrices.

4.6.4.2. Systems. The first system used in the experiment is HAC using GMMs system described in Section 4.6.1.

The second system is soft speaker clustering using GMMs. In general, soft speaker clustering methods used in this study first assign posterior probabilities of speakers given a speech segment. Using the weighted statistic, new statistics are synthesized as in Equation 4.15 for soft speaker clustering using GMMs. Although the Equation 4.15 uses centered statistics, non-centered statistics are synthesized for the soft speaker clustering using GMMs case. The adaptation coefficient is calculated as in Equation 2.12 using the synthesized zero order statistic. New mean vector is estimated as in the second line of the Equation 2.11 using the UBMs parameter. UBM is obtained by EM training of whole news bulletin. Relevance factor is same as the case HAC using GMMs system. After the speaker models are adapted using UBMs parameter, namely mean vectors, and synthesized statistics, posterior probabilities of speakers are updated for each speech segments. The initial posterior probabilities can be assigned randomly or result of a HAC system can be used to determine initial probabilities. Also, the number of speakers should be given for soft speaker clustering.

The other systems are soft speaker clustering using JFA and I-vectors. For both of these systems, the speaker model parameters are calculated as in the Equation 4.16 using the mean centered synthesized statistics. In the soft speaker clustering using JFA case, Equation 4.12 is used to obtain posterior probabilities of speaker for a given speech segment. In order to calculate the posterior using Equation 4.12, second order

statistics are also extracted. For the I-vectors case, the cosine distance of WCCN/LDA projected two I-vectors (speaker and segment) are used to obtain posterior probabilities. The cosine distance can not be used directly to obtain posterior probabilities. It is exponentiated and then normalized to have a posterior score.

Finally, HAC systems using JFA and I-vectors algorithms are also established for the reasons described in the Section 4.6.4.3. In these systems, model parameters and statistics are extracted using each speech segment. Cross likelihood ratio is obtained as in Equation 4.5. In these systems, UBM in Equation 4.5 is not a GMM but an I-vector or a JFA model which is modeled using the whole news bulletin for the JFA and I-vectors algorithms, respectively.

4.6.4.3. Results for More Than Two Speakers Segmentation. The new bulletins contain several speakers. Typically, there are 3 to 9 speakers in the test samples. It is experimentally observed that soft speaker clustering yielded higher error rates than HAC based systems for the news bulletins containing several number of speakers. The soft speaker clustering systems are initialized randomly and by using output of a HAC based system, however both approaches resulted in poor system performances, as well. It is deduced that soft clustering approaches are not suitable for the test samples containing several speakers. Therefore, only HAC based systems are reported in this section and performance of soft clustering approaches are reported in Section 4.6.4.4 where there are two speakers in all samples.

The performance of GMM, JFA and I-vector based HAC systems are given Table 4.4. The best results are obtained with JFA case. When number of speakers are high, I-vector case performed worse than GMM case but performed better than GMM case when number of speakers are low. Viterbi refinement did not reduce the error rates significantly, thus they are not reported here.

It can be deduced from the results that HAC using JFA is optimal in segmentation of news bulletins containing several speakers.

Table 4.4. Detection error rates of HAC systems (%).

# of Speakers	GMM	JFA	I-Vector
9	17.3	9.5	18.9
9	6.4	11.5	16.8
8	9.3	9.5	11.1
7	17.2	18.1	18.6
6	15.4	15.6	9.1
6	5.3	0.2	5.3
6	11.4	8.6	9.45
4	4	0	1.5
4	4.1	1.2	0
4	5.4	4.9	5.4
3	3	1.1	2.3
3	1.9	0	0
Average	8.4	6.7	8.2

4.6.4.4. Results for Two Speakers Segmentation. In this part of the experiments, the test samples containing only two speakers are considered. However, there are no two speaker news bulletins in the test and training data. For this reason, test samples containing two speakers are generated using the test samples by selecting the two speakers who have longest amounts of speech segments in that sample. Soft speaker segmentation and HAC based systems yielded very low error rates, nearly zero. Comparison between these systems are not easy with these generated samples. Instead of using long test samples, around 40 different two speaker speaking test samples, which contain around 4 minutes of speech data, are generated.

When the data is short, the HAC systems performed poorly. Any of these systems yielded around 30% error rates. Therefore, only GMM, JFA and I-Vector based soft speaker segmentation results are presented in Table 4.5. The final 10 iterations of

Table 4.5. Detection error rates of soft speaker segmentation systems.

(%)	GMM	JFA	I-Vector
Average	12.3	10.6	10.4
Variance	20.2	5.6	5.2

viterbi refinement reduced the error rates drastically. The corresponding results are presented in the Table 4.6. The JFA and I-Vector cases performed similarly. This

Table 4.6. Detection error rates of soft speaker segmentation systems after Viterbi.

(%)	GMM	JFA	I-Vector
Average	1.4	1.1	1.0
Variance	2.6	1.6	1.3

may be caused by usage of the same total variability matrix by these two systems. Soft speaker segmentation system using GMMs has worse performance rates than the systems using JFA and I-vectors. It can be deduced from the results that soft speaker clustering using I-Vectors is optimal in segmentation of short durational speech samples containing two speakers.

5. CONCLUSION

In this study, two main subjects using speaker related information, speaker verification and audio diarization are studied. The study in the speaker verification task is divided into two parts. In the first part, two baseline systems, GMM/UBM and GMM/SVM methodologies, are developed for NIST SRE 2010 submission. It is observed that the relatively newer approach GMM/SVM outperformed the earliest approach GMM/UBM. It is also observed that these two baseline systems performed worse than state-of-the-art systems most of which were based on factor analysis approach. However, relatively good and acceptable results were obtained for the first time submission to NIST SRE. A valuable research and programing experience had gained during preparation period to this evaluation. No special efforts for different microphone or telephone condition in training and test data are considered but only right balance between microphone and telephone data is considered as the development data. For this reason, only core condition 1 and 5, only microphone and telephone data in training and test conditions, are analyzed for considering system performances. GMM/SVM approach yielded 9% and 25% error reduction in EER than GMM/UBM approach. A third method is obtained by a fusion strategy of these two baseline system. The performance of the fusion strategy is in between those two systems because the fusion parameters are not estimated scores that are generated by these two systems on a common test set. It is observed that actual DCF values are much higher than minimum achievable DCF. This means that calibration of the systems are not well established because of inappropriate selection of decision threshold. A better approach could be to determine threshold value by testing the system under large set of validation data. In order to obtain robust systems to environmental condition, channel compensation techniques and score normalization should be considered, as well.

On the second part of the study in speaker verification, problem with training with limited amount of microphone data is considered. The system used in this part is a state-of-the-art I-vector approach. Microphone suitable approach in training total variability space had been offered. In this study, a correct configuration of data in

PLDA training on I-vector level performed better results than classical PLDA training. The following can be deduced from this study. Most of the speaker variability is captured by total variability space that is trained using large set of telephone data. The residual speaker variability is expected to be captured by total variability space of the microphone condition, however it also captures most of channel variabilities. Therefore, addition of further channel space to PLDA model can better account for microphone conditions on I-vector level. This approach is consistently confirmed with the experiments related to all microphone conditions in NIST SRE 2010. This approach is similar to I-vector extractor training, first all system is trained using telephone data and further additions are made using only microphone data. Therefore, the proposed approach is consistent way of training with microphone data.

Study in audio diarization is divided into three parts. In the first part, speaker adaptation via MLLR is performed using audio diarization information. It is known that performance of LVCSR systems can be improved by speaker adaptation. This is confirmed with the tests with Turkish Broadcast News database under different training data conditions of the speech recognition system. When speaker adaptation is applied 3% absolute WER reduction is obtained with the best case compared to the speech recognition case with speaker independent models. In this case, WER that can be obtained using perfect diarization, i.e. manually labeled audio, is nearly achieved with automatic diarization. Therefore, the gain in WER that can be obtained from a better diarization system is very limited.

In the second part of the study in audio diarization is concerned with detection of simultaneous speakers. A novel approach based on the baseline audio diarization system is proposed. It uses a Maximum Likelihood approach to decompose features into two samples for the given speaker models. In the experiment, detection is also performed with no decomposition to test the validity of the proposed approach. It is shown in the experiments that the proposed approach performed better result in the overall error rates. Although missed speaker and speaker errors in the proposed system are slightly higher than the comparison system, the increased error in false alarm rate for the proposed system is much lower. This resulted in lower overall

error rate for the proposed approach. It would be interesting to analyze detection of simultaneous speakers using a soft clustering approach because there is no hard assignment to segments in this method. The proposed decomposition may perform better results in soft clustering approach and this can be a future direction for the proposed approach.

In the last part of the study, comparison between factor analysis based and baseline speaker segmentation systems are presented. HAC and soft speaker segmentation systems using GMMs, JFA and I-Vectors are tested under different samples with various durations and number of speakers. HAC based segmentation systems performed well when number of speaker and duration of the test samples are high and soft speaker segmentation systems performed well when number of speaker and duration of the test samples are low. In both of these cases, factor analysis based systems yielded lower amounts of error rates than the baseline GMMs based system.

APPENDIX A: DETAILED RESULTS OF MICROPHONE CONDITIONS

Table A.1. Results of Tel400 I-vector scoring methods for condition 1.

Method	Data	EER	minDCF	minDCF_old
Plda	Tel	0.0329	0.4908	0.1535
	Tel + Int+Mic	0.0307	0.46	0.1388
	Int+Mic	0.0446	0.5529	0.2019
Plda C. Comp	tel:tel	0.0346	0.5331	0.1634
	tel:tel+int+mic	**0.0271**	0.442	0.1318
	tel:mic+int	0.0297	0.4292	**0.1311**
	tel+int+mic:tel+mic+int	0.0279	0.4724	0.1419
	tel+int+mic:tel	0.0325	0.4837	0.1497
	tel+int+mic:mic+int	0.0303	0.442	0.1423
	int+mic:int+mic	0.0424	0.5252	0.1898
LDA	Tel	0.0415	0.4875	0.1904
	Tel Snorm	0.0351	0.6396	0.1865
	Tel + Int+Mic	0.0347	**0.4288**	0.1594
	Tel + Int+Mic Snorm	0.0302	0.5496	0.1536
	Int+Mic	0.0482	0.5955	0.2216
	Int+Mic Snorm	0.0383	0.5628	0.1735

Table A.2. Results of Tel400Int200 I-vector scoring methods for condition 1.

Method	Data	EER	minDCF	minDCF_old
Plda	Tel	0.0383	0.6093	0.1952
	Tel + Int+Mic	0.0424	0.6368	0.2057
	Int+Mic	0.0559	0.6244	0.2559
Plda C. Comp	tel:tel	0.041	0.6706	0.2076
	tel:tel+int+mic	0.0284	0.3882	**0.1101**
	tel:mic+int	**0.0274**	**0.3572**	0.1192
	tel+int+mic:tel+mic+int	0.0302	0.5197	0.1461
	tel+int+mic:tel	0.0522	0.7428	0.2538
	tel+int+mic:mic+int	0.0297	0.4543	0.1345
	int+mic:int+mic	0.0496	0.575	0.2272
LDA	Tel	0.0658	0.7429	0.3176
	Tel Snorm	0.056	0.6871	0.2358
	Tel + Int+Mic	0.0492	0.5332	0.2163
	Tel + Int+Mic Snorm	0.0451	0.6238	0.2038
	Int+Mic	0.0514	0.5849	0.2249
	Int+Mic Snorm	0.0401	0.5628	0.1823

Table A.3. Results of Tel400Tel200 I-vector scoring methods for condition 1.

Method	Data	EER	minDCF	minDCF_old
Plda	Tel	0.0306	0.4835	0.1594
	Tel + Int+Mic	0.0297	0.42	**0.1319**
	Int+Mic	0.0523	0.6357	0.2377
Plda C. Comp	tel:tel	0.0311	0.5129	0.1575
	tel:tel+int+mic	**0.0266**	0.4148	0.1374
	tel:mic+int	0.0288	**0.3982**	0.1392
	tel+int+mic:tel+mic+int	0.0274	0.4168	0.1455
	tel+int+mic:tel	0.0297	0.4485	0.1369
	tel+int+mic:mic+int	0.0293	0.4192	0.15
	int+mic:int+mic	0.0527	0.6027	0.2447
LDA	Tel	0.046	0.5456	0.2286
	Tel Snorm	0.0419	0.618	0.2072
	Tel + Int+Mic	0.0428	0.5029	0.1902
	Tel + Int+Mic Snorm	0.0387	0.5376	0.1779
	Int+Mic	0.0545	0.6245	0.2457
	Int+Mic Snorm	0.0451	0.6313	0.2032

Table A.4. Results of Tel400 I-vector scoring methods for condition 2.

Method	Data	EER	minDCF	minDCF_old
Plda	Tel	0.063	0.7858	0.3199
	Tel + Int+Mic	0.0516	0.7087	0.2631
	Int+Mic	0.0695	0.7684	0.331
Plda C. Comp	tel:tel	0.0629	0.7993	0.3219
	tel:tel+int+mic	**0.0431**	0.6322	**0.2192**
	tel:mic+int	0.0462	**0.6155**	0.2221
	tel+int+mic:tel+mic+int	0.0445	0.6422	0.2242
	tel+int+mic:tel	0.0513	0.7267	0.2708
	tel+int+mic:mic+int	0.0477	0.6397	0.2303
	int+mic:int+mic	0.062	0.7085	0.2985
LDA	Tel	0.08	0.7961	0.3688
	Tel Snorm	0.0732	0.8235	0.3475
	Tel + Int+Mic	0.0652	0.7071	0.2932
	Tel + Int+Mic Snorm	0.0573	0.7413	0.2748
	Int+Mic	0.0809	0.7769	0.3601
	Int+Mic Snorm	0.0605	0.7412	0.2789

Table A.5. Results of Tel400Int200 I-vector scoring methods for condition 2.

Method	Data	EER	minDCF	minDCF_old
Plda	Tel	0.0696	0.8219	0.3463
	Tel + Int+Mic	0.0752	0.8325	0.354
	Int+Mic	0.0887	0.8509	0.4057
Plda C. Comp	tel:tel	0.0701	0.8356	0.3526
	tel:tel+int+mic	**0.043**	0.6234	**0.2114**
	tel:mic+int	0.0474	**0.6078**	0.2187
	tel+int+mic:tel+mic+int	0.0484	0.6859	0.2455
	tel+int+mic:tel	0.0883	0.9039	0.4149
	tel+int+mic:mic+int	0.0488	0.6551	0.2405
	int+mic:int+mic	0.0804	0.7923	0.361
LDA	Tel	0.1202	0.9039	0.5237
	Tel Snorm	0.1071	0.8737	0.4606
	Tel + Int+Mic	0.085	0.7775	0.378
	Tel + Int+Mic Snorm	0.0736	0.8114	0.3345
	Int+Mic	0.0796	0.7666	0.3542
	Int+Mic Snorm	0.0624	0.7496	0.2961

Table A.6. Results of Tel400Tel200 I-vector scoring methods for condition 2.

Method	Data	EER	minDCF	minDCF_old
Plda	Tel	0.0643	0.7764	0.3223
	Tel + Int+Mic	0.0529	0.6836	0.2574
	Int+Mic	0.08	0.8245	0.3794
Plda C. Comp	tel:tel	0.0604	0.754	0.3107
	tel:tel+int+mic	**0.0457**	0.6366	**0.2238**
	tel:mic+int	0.0497	**0.6328**	0.2299
	tel+int+mic:tel+mic+int	0.0463	0.6442	0.2296
	tel+int+mic:tel	0.0504	0.6929	0.2581
	tel+int+mic:mic+int	0.0514	0.6438	0.2398
	int+mic:int+mic	0.0788	0.8025	0.3781
LDA	Tel	0.0937	0.8186	0.4227
	Tel Snorm	0.0834	0.8368	0.3866
	Tel + Int+Mic	0.0749	0.7409	0.3371
	Tel + Int+Mic Snorm	0.0634	0.7722	0.3052
	Int+Mic	0.0853	0.8038	0.3903
	Int+Mic Snorm	0.0674	0.7701	0.3166

Table A.7. Results of Tel400 I-vector scoring methods for condition 3.

Method	Data	EER	minDCF	minDCF_old
Plda	Tel	0.0404	0.6463	0.2279
	Tel + Int+Mic	0.0342	0.6616	0.1938
	Int+Mic	0.0518	0.8022	0.2846
Plda C. Comp	tel:tel	0.0445	0.6717	0.2252
	tel:tel+int+mic	**0.0313**	0.597	**0.1676**
	tel:mic+int	0.0328	**0.5609**	0.1757
	tel+int+mic:tel+mic+int	0.0328	0.6026	0.171
	tel+int+mic:tel	0.0379	0.6615	0.1951
	tel+int+mic:mic+int	0.0354	0.5898	0.1859
	int+mic:int+mic	0.0522	0.7407	0.2731
LDA	Tel	0.0512	0.6981	0.2611
	Tel Snorm	0.0414	0.6961	0.2328
	Tel + Int+Mic	0.0451	0.6414	0.2278
	Tel + Int+Mic Snorm	0.0354	0.6277	0.1974
	Int+Mic	0.0645	0.7668	0.311
	Int+Mic Snorm	0.0472	0.6974	0.2291

Table A.8. Results of Tel400Int200 I-vector scoring methods for condition 3.

Method	Data	EER	minDCF	minDCF_old
Plda	Tel	0.0434	0.6886	0.2266
	Tel + Int+Mic	0.0589	0.7728	0.3075
	Int+Mic	0.0666	0.8155	0.3336
Plda C. Comp	tel:tel	0.043	0.7021	0.2306
	tel:tel+int+mic	**0.0304**	**0.5405**	**0.1502**
	tel:mic+int	0.0328	0.5534	0.163
	tel+int+mic:tel+mic+int	0.0317	0.6101	0.1769
	tel+int+mic:tel	0.0739	0.8227	0.3793
	tel+int+mic:mic+int	0.0343	0.5963	0.1895
	int+mic:int+mic	0.0651	0.7645	0.3083
LDA	Tel	0.0728	0.7831	0.341
	Tel Snorm	0.0671	0.7777	0.3086
	Tel + Int+Mic	0.0538	0.6967	0.2689
	Tel + Int+Mic Snorm	0.0477	0.6807	0.2391
	Int+Mic	0.0666	0.7495	0.3049
	Int+Mic Snorm	0.0518	0.6647	0.2421

Table A.9. Results of Tel400Tel200 I-vector scoring methods for condition 3.

Method	Data	EER	minDCF	minDCF_old
Plda	Tel	0.0395	0.65	0.2064
	Tel + Int+Mic	0.0328	0.6266	0.1867
	Int+Mic	0.06	0.8169	0.3071
Plda C. Comp	tel:tel	0.0373	0.63	0.2057
	tel:tel+int+mic	0.0302	0.6	**0.1708**
	tel:mic+int	0.0343	0.6053	0.1826
	tel+int+mic:tel+mic+int	**0.0297**	**0.5939**	0.173
	tel+int+mic:tel	**0.0297**	0.596	0.1846
	tel+int+mic:mic+int	0.0358	0.6337	0.1915
	int+mic:int+mic	0.0626	0.7972	0.3186
LDA	Tel	0.0538	0.6944	0.2602
	Tel Snorm	0.0441	0.7055	0.2321
	Tel + Int+Mic	0.0497	0.6765	0.2399
	Tel + Int+Mic Snorm	0.04	0.6543	0.2027
	Int+Mic	0.0723	0.7788	0.3321
	Int+Mic Snorm	0.0543	0.7202	0.2581

Table A.10. Results of Tel400 I-vector scoring methods for condition 4.

Method	Data	EER	minDCF	minDCF_old
Plda	Tel	0.0297	0.5648	0.1712
	Tel + Int+Mic	0.0257	0.473	0.1406
	Int+Mic	0.0428	0.6254	0.2115
Plda C. Comp	tel:tel	0.0298	0.5787	0.1742
	tel:tel+int+mic	**0.0219**	0.4468	**0.1122**
	tel:mic+int	0.022	**0.4244**	0.1173
	tel+int+mic:tel+mic+int	0.0243	0.4728	0.1199
	tel+int+mic:tel	0.0274	0.4994	0.1445
	tel+int+mic:mic+int	0.025	0.4418	0.1327
	int+mic:int+mic	0.0381	0.5637	0.1963
LDA	Tel	0.0452	0.5958	0.2231
	Tel Snorm	0.041	0.7185	0.2284
	Tel + Int+Mic	0.0345	0.5063	0.1625
	Tel + Int+Mic Snorm	0.0321	0.6491	0.1666
	Int+Mic	0.0447	0.608	0.2095
	Int+Mic Snorm	0.0345	0.6327	0.1841

Table A.11. Results of Tel400Int200 I-vector scoring methods for condition 4.

Method	Data	EER	minDCF	minDCF_old
Plda	Tel	0.0351	0.6304	0.2033
	Tel + Int+Mic	0.0375	0.6012	0.2061
	Int+Mic	0.0529	0.6673	0.248
Plda C. Comp	tel:tel	0.0363	0.6755	0.2074
	tel:tel+int+mic	**0.0214**	0.4187	**0.11**
	tel:mic+int	0.0226	**0.3939**	0.1134
	tel+int+mic:tel+mic+int	0.0238	0.5108	0.1368
	tel+int+mic:tel	0.0477	0.7272	0.258
	tel+int+mic:mic+int	0.025	0.4406	0.1331
	int+mic:int+mic	0.0498	0.6014	0.2233
LDA	Tel	0.0726	0.767	0.3325
	Tel Snorm	0.0596	0.7587	0.2821
	Tel + Int+Mic	0.0482	0.6051	0.2177
	Tel + Int+Mic Snorm	0.044	0.6934	0.1998
	Int+Mic	0.0464	0.5924	0.2112
	Int+Mic Snorm	0.0364	0.6091	0.1778

Table A.12. Results of Tel400Tel200 I-vector scoring methods for condition 4.

Method	Data	EER	minDCF	minDCF_old
Plda	Tel	0.0309	0.5601	0.1747
	Tel + Int+Mic	0.0237	0.4422	0.1351
	Int+Mic	0.047	0.637	0.2322
Plda C. Comp	tel:tel	0.0291	0.5706	0.1721
	tel:tel+int+mic	0.0219	0.4475	**0.112**
	tel:mic+int	0.022	**0.4152**	0.1195
	tel+int+mic:tel+mic+int	**0.0214**	0.4581	0.1201
	tel+int+mic:tel	0.0232	0.4765	0.138
	tel+int+mic:mic+int	0.0237	0.4265	0.1292
	int+mic:int+mic	0.048	0.6577	0.242
LDA	Tel	0.0535	0.6462	0.264
	Tel Snorm	0.047	0.7284	0.2531
	Tel + Int+Mic	0.041	0.5581	0.1971
	Tel + Int+Mic Snorm	0.0351	0.6652	0.1904
	Int+Mic	0.0518	0.6314	0.2434
	Int+Mic Snorm	0.0405	0.6793	0.1998

Table A.13. Results of Tel400 I-vector scoring methods for condition 5.

Method	Data	EER	minDCF	minDCF_old
Plda	Tel	0.0292	0.4691	0.1389
	Tel + Int+Mic	0.0295	**0.4683**	0.148
	Int+Mic	0.0549	0.6909	0.2559
Plda C. Comp	tel:tel	0.027	0.4782	**0.1378**
	tel:tel+int+mic	0.0278	0.5317	0.1471
	tel:mic+int	0.0327	0.5232	0.1589
	tel+int+mic:tel+mic+int	0.0292	0.5484	0.1539
	tel+int+mic:tel	**0.0268**	0.4794	0.1439
	tel+int+mic:mic+int	0.0335	0.5368	0.1661
	int+mic:int+mic	0.0535	0.6787	0.2585
LDA	Tel	0.0429	0.5718	0.1987
	Tel Snorm	0.0337	0.5202	0.1738
	Tel + Int+Mic	0.0443	0.569	0.2091
	Tel + Int+Mic Snorm	0.0351	0.5255	0.1811
	Int+Mic	0.0732	0.7435	0.3184
	Int+Mic Snorm	0.0629	0.7246	0.281

Table A.14. Results of Tel400Int200 I-vector scoring methods for condition 5.

Method	Data	EER	minDCF	minDCF_old
Plda	Tel	0.0306	0.4897	0.1459
	Tel + Int+Mic	0.0302	0.4996	0.1607
	Int+Mic	0.0651	0.7367	0.2838
Plda C. Comp	tel:tel	0.0289	**0.4724**	0.1437
	tel:tel+int+mic	0.0289	0.4896	**0.1352**
	tel:mic+int	0.0316	0.4911	0.1495
	tel+int+mic:tel+mic+int	**0.0278**	0.5081	0.1447
	tel+int+mic:tel	0.0321	0.5076	0.167
	tel+int+mic:mic+int	0.0292	0.4918	0.1515
	int+mic:int+mic	0.0634	0.6801	0.2727
LDA	Tel	0.0499	0.5873	0.2222
	Tel Snorm	0.0391	0.5248	0.1868
	Tel + Int+Mic	0.0464	0.5646	0.214
	Tel + Int+Mic Snorm	0.037	0.5251	0.1889
	Int+Mic	0.0721	0.7497	0.3193
	Int+Mic Snorm	0.0653	0.717	0.2941

Table A.15. Results of Tel400Tel200 I-vector scoring methods for condition 5.

Method	Data	EER	minDCF	minDCF_old
Plda	Tel	0.0275	0.4458	0.1382
	Tel + Int+Mic	0.0294	0.4365	0.141
	Int+Mic	0.0592	0.7086	0.2757
Plda C. Comp	tel:tel	**0.0254**	0.4441	**0.1323**
	tel:tel+int+mic	0.0286	0.5275	0.1466
	tel:mic+int	0.0335	0.5254	0.1617
	tel+int+mic:tel+mic+int	0.03	0.5122	0.1488
	tel+int+mic:tel	0.027	**0.425**	0.1337
	tel+int+mic:mic+int	0.0351	0.5285	0.1674
	int+mic:int+mic	0.0621	0.7385	0.3002
LDA	Tel	0.0448	0.5692	0.2018
	Tel Snorm	0.0346	0.5001	0.173
	Tel + Int+Mic	0.0454	0.5617	0.2058
	Tel + Int+Mic Snorm	0.0354	0.5354	0.1756
	Int+Mic	0.075	0.7723	0.3414
	Int+Mic Snorm	0.0632	0.7216	0.2846

APPENDIX B: RESULTS OF BOUN SUBMISSION FOR NIST SRE 2010

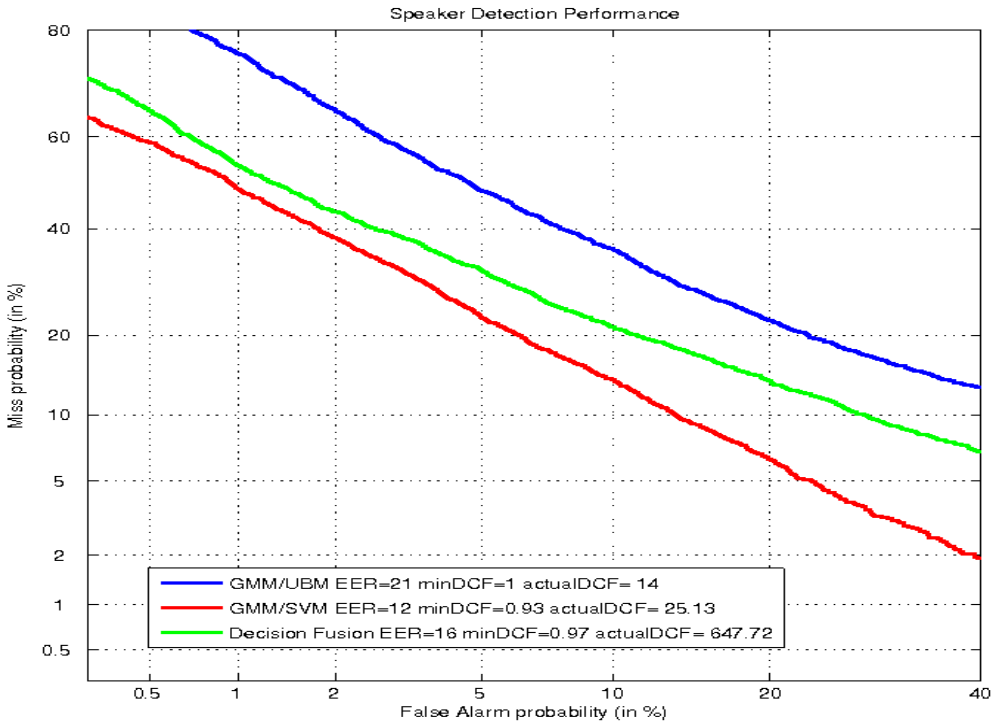

Figure B.1. Results for core-core condition 2.

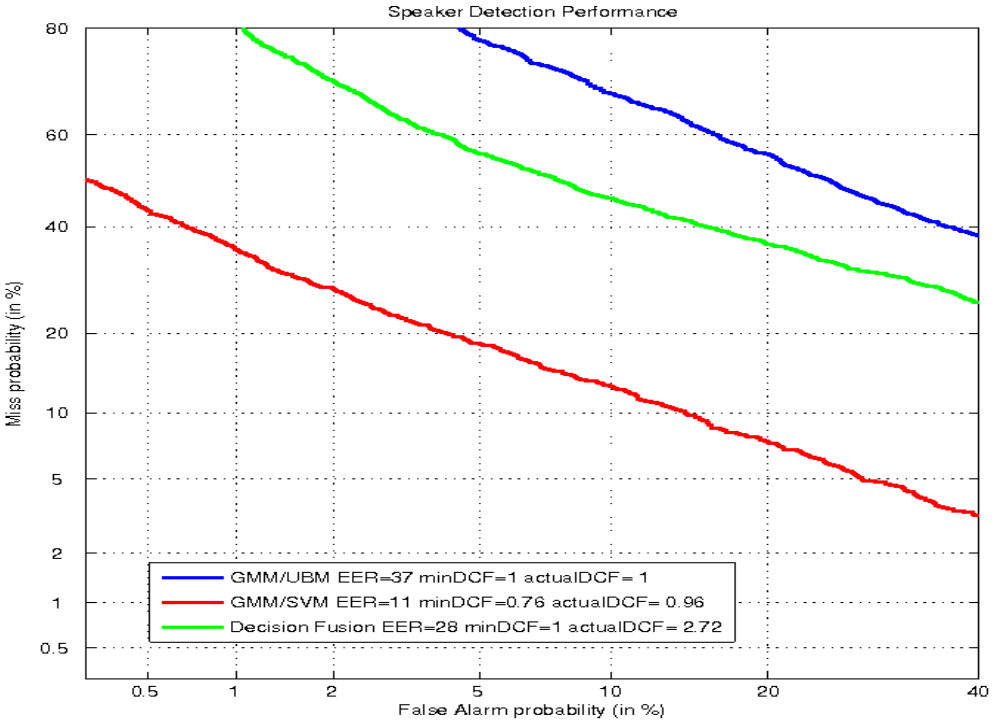

Figure B.2. Results for core-core condition 3.

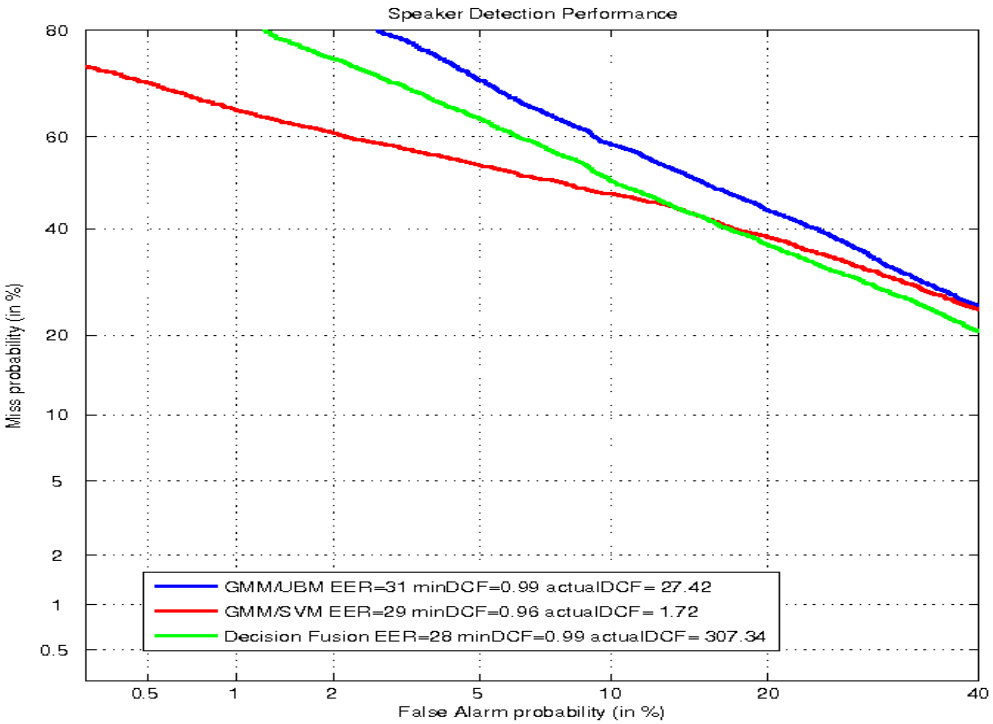

Figure B.3. Results for core-core condition 4.

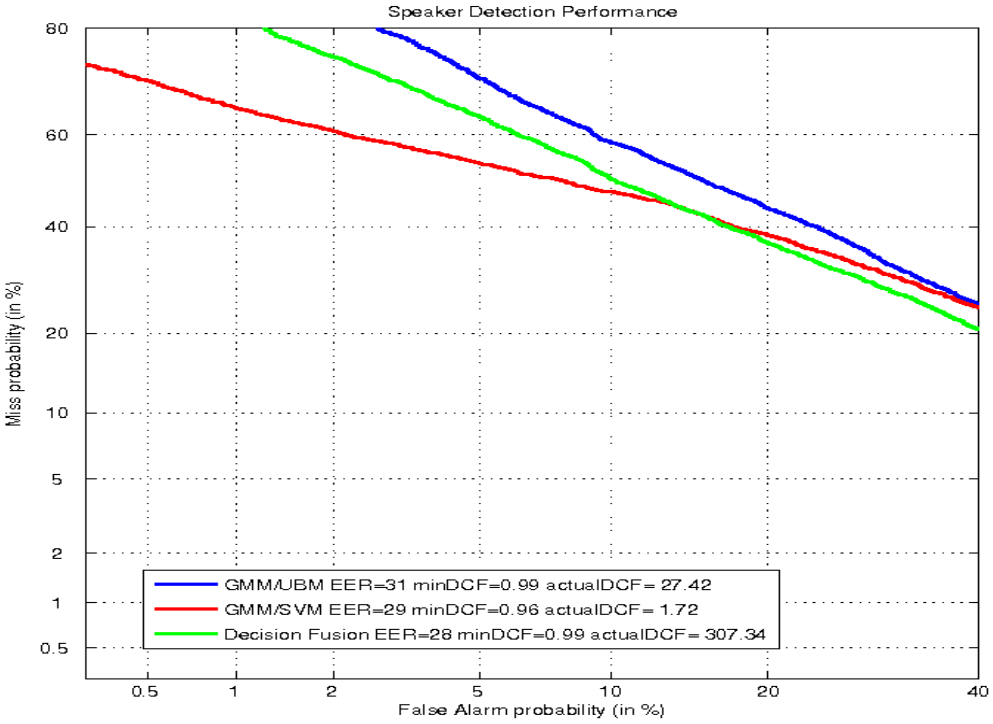

Figure B.4. Results for core-core condition 6.

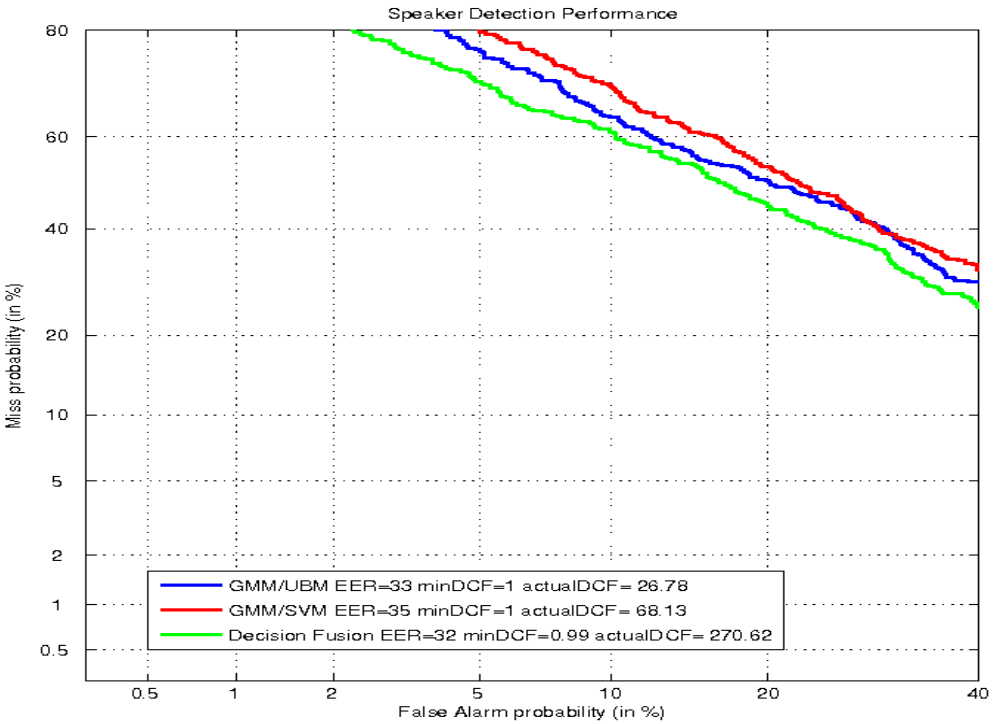

Figure B.5. Results for core-core condition 7.

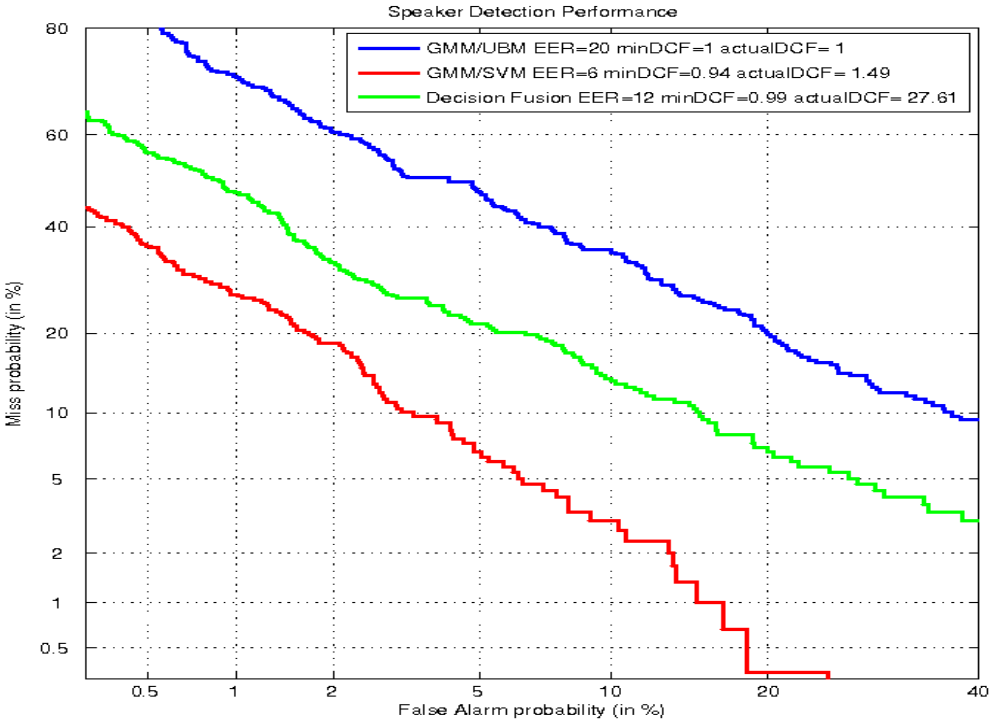

Figure B.6. Results for core-core condition 8.

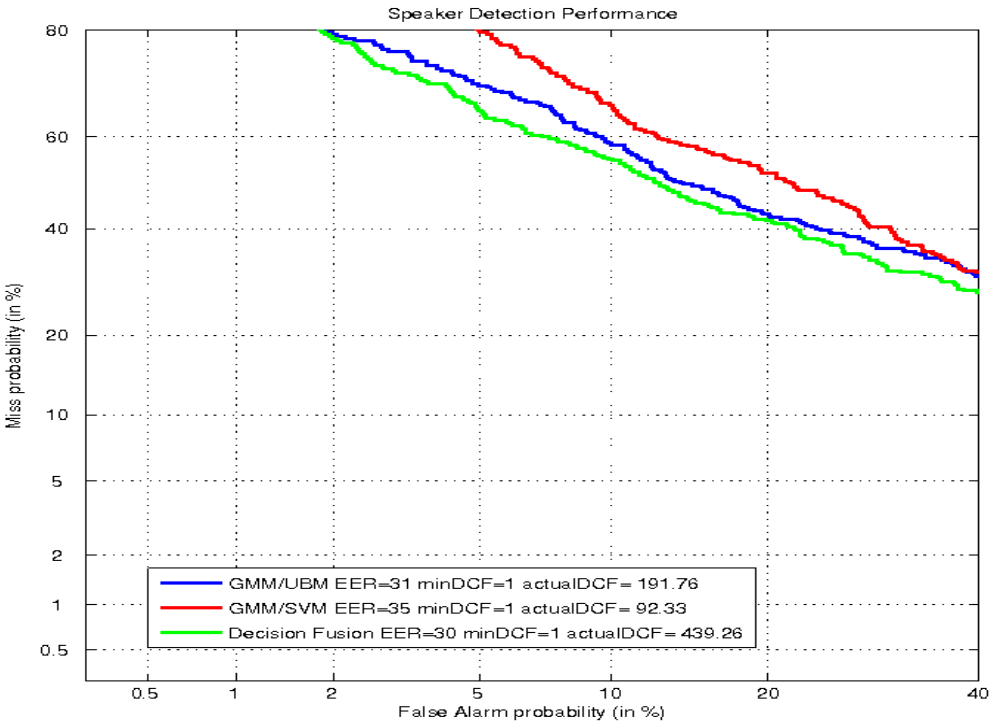

Figure B.7. Results for core-core condition 9.

REFERENCES

1. Ganchev, T., N. Fakotakis and G. Kokkinakis, "Comparative Evaluation of Various MFCC Implementations on the Speaker Verification Task", *Proceedings of 10th International Conference on Speech and Computer*, Vol. 2, pp. 191–194, 2005.

2. Hermasnsky, H., "Perceptual Linear Predictive Analysis of Speech", *Journal of the Acoustical Society of America*, Vol. 87, pp. 1738–1752, 1990.

3. Bimbot, F., J. F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, et al., "A Tutorial on Text-Independent Speaker Verification", *EURASIP Journal on Applied Signal Processing*, Vol. 4, pp. 430–451, 2004.

4. Doddington, G., M. A. Przybocki, A. Martin and D. A. Reynolds, "The NIST Speaker Recognition Evaluation - Overview, Methodology, Systems, Results, Perspective", *Speech Communication*, Vol. 2, pp. 263–286, 2000.

5. Barras, C. and J.L. Gauvain, "Feature and Score Normalization for Speaker Verification of Cellular Data", *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 2, pp. 49–52, 2003.

6. Burget, L., P. Matějka, P. Schwarz, O. Glembek and J. Cernocký, "Analysis of Feature Extraction and Channel Compensation in a GMM Speaker Recognition System", *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 15, No. 7, pp. 1979–1986, 2007.

7. Garcia, A. A. and R. J. Mammone, "Channel-Robust Speaker Identification Using Modified Mean Cepstral Mean Normalization with Frequency Warping", *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 1, pp. 325–328, 1999.

8. Pelecanos, J. and S. Sridharan, "Feature Warping for Robust Speaker Verification",

Proceedings of Odyssey: The Speaker Recognition Workshop, Vol. 1, pp. 213–218, 2001.

9. Xiang, B., U. V. Chaudhari, G. N. Ramaswamy and R. A. Gopinath, "Short-Time Gaussianization for Robust Speaker Verification", *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 1, pp. 681–684, 2002.

10. Kumar, N., *Investigation of Silicon-Auditory Models and Generalization of Linear Discriminant Analysis for Improved Speech Recognition*, Ph.D. Dissertation, Johns Hopkins University, 1997.

11. Reynolds, D. A., T. F. Quatieri and R. B. Dunn, "Speaker Verification Using Adapted Gaussian Mixture Models", *Digital Signal Processing*, Vol. 10, No. 1–3, pp. 19–41, 2000.

12. Reynolds, D. A., "Comparison of Background Normalization Methods for Text-Independent Speaker Verification", *Proceedings of 5th European Conference on Speech Communication and Technology*, Vol. 2, pp. 963–966, 1997.

13. Reynolds, D. A., "Universal Background Models", *Encyclopedia of Biometric Recognition*, Vol. 1, pp. 659–663, 2008.

14. Hasan, T. and J. H. Hansen, "A Study on Universal Background Model Training in Speaker Verification", *IEEE Transactions on Audio, Speech and Language Processing*, in submission.

15. Lee, C. H., C. H. Lin and B.-H. Juang, "A Study on Speaker Adaptation of the Parameters of Continuous Density Hidden Markov Models", *IEEE Transactions on Signal Processing*, Vol. 39, No. 4, pp. 806–814, 1991.

16. Leggetter, C. J. and P. C. Woodland, "Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models", *Computer*

Speech and Language, Vol. 9, pp. 171–185, 1995.

17. Campbell, W. M., D. E. Sturim, D. A. Reynolds, "Support Vector Machines Using GMM Supervectors for Speaker Verification", *IEEE Signal Processing Letters*, Vol. 13, No. 5, pp. 308–311, 2006.

18. Campbell, W. M., J. P. Campbell, D. A. Reynolds, E. Singer, P.A. Torres-Carrasquillo, "Support Vector Machines for Speaker and Language Recognition", *Computer Speech and Language*, Vol. 20, pp. 210–229, 2006.

19. Wan, V. and S. Renals, "SVMSVM: Support Vector Machine Speaker Verification methodology", *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 2, pp. 221–224, 2003.

20. Kenny, P., G. Boulianne, P. Ouellet and P. Dumouchel, "Speaker and Session Variability in GMM-Based Speaker Verification", *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 15, pp. 1448–1460, 2007.

21. Kenny, P., G. Boulianne, P. Ouellet and P. Dumouchel, "Joint Factor Analysis Versus Eigenchannels in Speaker Recognition" *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 15, pp. 1435–1447, 2007.

22. Dehak, N., P. Kenny and P. Dumouchel, "Modeling Prosodic Features with Joint Factor Analysis for Speaker Verification", *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 15, pp. 2095–2103, 2007.

23. Reynolds, D. A., "Channel Robust Speaker Verification via Feature Mapping", *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 2, pp. 53–56, 2003.

24. Kenny, P., G. Boulianne and P. Dumouchel, "Eigenvoice Modeling with Sparse Training Data", *IEEE Transactions on Speech and Audio Processing*, Vol. 13, pp. 345–359, 2005.

25. Vogt, R., B. Baker and S. Sridharan, "Modelling Session Variability in Text-Independent Speaker Verification", *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 1, pp. 1–4, 2006.

26. Kenny, P., "Joint Factor Analysis of Speaker and Session Variability: Theory and Algorithms", Technical Report CRIM-06/08-13, 2005.

27. Kenny, P., G. Boulianne, P. Ouellet and P. Dumouchel, "Speaker Adaptation Using an Eigenphone Basis", *IEEE Transactions on Audio Speech and Signal Processing*, Vol. 12, No. 6, pp. 579–589, 2004.

28. Dehak, N., *Discriminative and Generative Approches for Long- and Short-Term Speaker Characteristics Modeling: Application to Speaker Verification*, Ph.D. Thesis, Ecole de Technologie Suprieure, 2009.

29. Dehak, N., R. Dehak, P. Kenny, N. Brümmer, P. Ouellet, et al., "Support Vector Machines versus Fast Scoring in the Low Dimensional Total Variability Space for Speaker Verification", *10th Annual Conference of the International Speech Communication Association*, Vol. 16, pp. 1559–1562, 2009.

30. Dehak, N., P. Kenny, R. Dehak, P. Dumouchel and P. Ouellet, "Front-End Factor Analysis for Speaker Verification", *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 19, pp. 788–798, 2011.

31. Brümmer, N., "The EM Algorithm and Minimum Divergence", Agnitio Labs Technical Report, 2009.

32. Dehak, N., P. Kenny, R. Dehak, O. Glembek, P. Dumouchel, et al., "Support Vector Machines and Joint Factor Analysis for Speaker Verification", *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 2, pp. 4237–4240, 2009.

33. Hatch A., S. Kajarekar and A. Stolcke, "Within Class Covariance Normalization for

SVM-Based Speaker Recognition", *International Conference on Spoken Language Processing*, Vol. 14, pp. 1471–1474, 2006.

34. Campbell, W.M., D. E. Sturim, D. A. Reynolds and A. Solomonoff, "SVM Based Speaker Verification Using a GMM Supervector Kernel and NAP Variability Compensation", *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 1, pp. 97–100, 2006.

35. Vogt R., S. Kajarekar and S. Sridharan, "Discriminant NAP for SVM Speaker Recognition", *Proceedings of Odyssey: The Speaker and Language Recognition Workshop*, Vol. 4, pp. 10–15, 2008.

36. Senoussaoui, M., P. Kenny, P. Dumouchel and F. Castaldo, "Well Calibrated Heavy Tailed Bayesian Speaker Verification for Microphone Speech", *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 4, pp. 4824–4827, 2011.

37. Brümmer, N., "Bayesian PLDA", 2010, `https://sites.google.com/site/nikobrummer/bplda.pdf`, accessed at May 2011.

38. Villalba, J. and N. Brümmer, "Bayesian Two Covariance Model Integrating out the Speaker Space Distribution", 2010, `http://sites.google.com/site/nikobrummer/bay2covmuB.pdf`, accessed at May 2011.

39. Dikici, E., *Effects of Data Duration, Model Size and Session Variability on Speaker Verification Performance*, MS Thesis, Boğaziçi University, 2008.

40. Senoussaoui, M., P. Kenny, N. Dehak and P. Dumouchel, "An I-vector Extractor Suitable for Speaker Recognition with Both Microphone and Telephone Speech", *Proceedings of Odyssey: Speker and Language Recognition Workshop*, Vol. 6, pp. 451–456, 2010.

41. Heck, L. and M. Weintraub, "Handset-Dependent Background Models for Robust

Text-Independent Speaker Verification", *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 2, pp. 1071–1074, 1997.

42. Rosenberg, A., J. Delong, C. Lee, B. Juang and F. Soong, "The Use of Cohort Normalized Scores for Speaker Recognition", *International Conference on Spoken Language Processing*, Vol. 3, pp. 599–602, 1992.

43. Li, K. P. and J. E. Porter, "Normalizations and Selection of Speech Segments for Speaker Recognition Scoring", *Proceedings of IEEE International Conference Acoustics, Speech, Signal Processing*, Vol. 1, pp. 595–598, 1988.

44. Auckenthaler, R., M. Carey and H. Lloyd-Thomas, "Score Normalization for Text-Independent Speaker Verification Systems", *Digital Signal Processing*, Vol. 10, pp. 42–54, 2000.

45. Brümmer, N. and E. de Villiers, "The Speaker Partitioning Problem", *Proceedings of Odyssey: Speaker and Language Recognition Workshop*, Vol. 4, pp. 194–201, 2010.

46. Zheng, R., S. Zhang and B. Xu, "A Comparative Study of Feature and Score Normalization for Speaker Verification", *Advances in Biometrics*, Vol. 3832, pp. 531–538, 2005

47. Vogt, R., B. Baker and S. Sridharan, "Modeling Session Variability in Text-Independent Speaker Verification", *9th European Conference on Speech Communication and Technology*, Vol. 9, pp. 3117–3120, 2005.

48. Kenny, P., P. Ouellet, N. Dehak, V. Gupta and P. Dumouchel, "A Study of Inter-Speaker Variability in Speaker Verification", *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 16, pp. 980–988, 2008.

49. Reynolds, D.A., "Comparison of Background Normalisation Methods for Text-Independent Speaker Verification", *5th European Conference on Speech Communi-*

cation and Technology, Vol. 2, pp. 963–966, 1997.

50. Swets, John A, ed., *Signal Detection and Recognition by Human Observers*, John Wiley & Sons, Inc., New York, pp. 611-648, 1964.

51. Martin, A., G. Doddington, T. Kamm, M. Ordowski and M. Przybocki, "The DET Curve in Assessment of Detection Task Performance", *5th European Conference on Speech Communication and Technology*, Vol. 4, pp. 1895–1898, 1997.

52. "The NIST Year 2010 Speaker Recognition Evaluation Plan", 2010, `http://www.itl.nist.gov/iad/mig/tests/sre/2010/index.html`, accessed at May 2011.

53. Brümmer, N., O. Glembek, P. Kenny, "The ABC and CRIM Systems for the NIST 2010 Speaker Recognition Evaluation", 2010, `http://www.crim.ca/perso/patrick.kenny/abc4sre_slides-joined.pdf`, accessed at May 2011.

54. Mayoue, A., "Reference System Based on Speech Modality BECARS/HTK", Technical report, GET-INT, 2008.

55. "HTK Speech Recognition Toolkit", 2009, `http://htk.eng.cam.ac.uk/`, accessed at May 2011.

56. "UNIANAL Universal Speech Analysis and Synthesis", 2004, `http://speech.fit.vutbr.cz/files/software/unianal/unianal.tar.gz`, accessed at May 2011.

57. Mokbel, C., H. Mokbel, R. Blouet, G. Aversano, "BECARS Library and Tools for Speaker Verification", 2008, `http://www.tsi.enst.fr/becars/index.php`, accessed at May 2011.

58. "SPro Speech Signal Processing Toolkit", 2004, `http://www.irisa.fr/metiss/guig/spro/`, accessed at May 2011.

59. "FIR Echo Canceller", 2007, `http://www.isip.piconepress.com/projects/`

speech/software/legacy/fir_echo_canceller/, accessed at May 2011.

60. Collobert, R. and S. Bengio, "SVMTorch: Support Vector Machines for Large-Scale Regression Problems", *Journal of Machine Learning Research*, Vol. 1, pp. 143–160, 2001.

61. Demir, C. and M. U. Doğan, "Speech-Music Segmentation System for Speech Recognition", *Signal Processing and Communications Applications, 2009. SIU 2009. IEEE 17th* , pp. 608-611, 2009.

62. Kenny, P., D. A. Reynolds and F. Castaldo, "Diarization of Telephone Conversations Using Factor Analysis", *IEEE Journal of Selected Topics in Signal Processing*, Vol. 4, pp. 1059–1070, 2010.

63. Arisoy E., D. Can, S. Parlak, H. Sak, M. Saraclar, "Turkish Broadcast News Transcription and Retrieval", *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 17, pp. 874–883, 2009.

64. Wooters, C., J. Fung, B. Peskin and X. Anguera, "Toward Robust Speaker Segmentation: The ICSI-SRI Fall 2004 Diarization System", *Rich Transcription Workshop*, Vol. 5, pp. 1212–1224, 2004.

65. Siegler, M. A., U. Jain, B. Raj and R. M. Stern, "Automatic Segmentation, Classification and Clustering of Broadcast News", *DARPA Speech Recognition Workshop*, Vol. 2, pp. 97–99, 1997.

66. Chen, S. S. and P. S. Gopalakrishnam, "Speaker, Environment and Channel Change Detection and CLustering via the Bayesian Information Criterion", *DARPA Broadcast News Transcription and Understanding Workshop*, Vol. 4, pp. 127–132, 1998.

67. Tranter, S. and D. A. Reynolds, "An Overview of Automatic Speaker Diarization Systems", *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 14,

pp. 1557–1565, 2006.

68. Kenny, P., "Bayesian Analysis of Speaker Diarization with Eigenvoice Priors", Technical Report, Montreal, CRIM, 2008

69. The National Institute of Standards and Technology, "The 2009 Rich Transcription Meeting Recognition Evaluation Plan", 2009, `http://www.itl.nist.gov/iad/mig/tests/rt/2009/docs/rt09-meeting-eval-plan-v2.pdf`, accessed at May 2011.