Speaker Diarization using GMM Mean Supervector and Advanced Dimensionality Reduction Algorithms

Nurit Spingarn

Speaker Diarization using GMM Mean Supervector and Advanced Dimensionality Reduction Algorithms

Research Thesis

As Partial Fulfillment of the Requirements for
the Degree Master of Science in Electrical Engineering

Nurit Spingarn

Submitted to the Senate of the Technion—Israel Institute of Technology

Tevet 5776 Haifa Januar 2016

The Research Thesis Was Done Under The Supervision Of Professor Israel Cohen In The Department Of Electrical Engineering.

Acknowledgments

I wish to express my deep gratitude and appreciation to Prof. Israel Cohen for his guidance and dedicated supervision. I appreciate his professional support and contribution to my personal development.
I would also like to thank SIPL stuff for their valuable support.
I would like to thank my friends and colleagues in the Technion. Special thanks to Saman mouzshada and Hadas Benisty for their support and friendship and for their helpful comments. I would like to express my deep gratitude to my dear family, my parents Nomi and Rafael and to my brother Omer, for their love, encouragement and support all the way. David, my husband, for his endless support, encouragement and powerful love, willing me to success and perfection.

The research was supported by The Israel Science Foundation (Grant no. 1130/11). The generous Financial help of the Technion is greatly Acknowledged.

List of Publications

1. N. Spingarn, S. Mousazadeh and I. Cohen "Voice Activity Detection in Transient Noise Environment Using Laplacian Pyramid Algorithm" Proc. 14th International Workshop on Acoustic Signal Enhancement, IWAENC-2014, Antibes Juan les Pins, French Riviera, Sep. 9 – 11, 2014.

2. N. Spingarn, S. Mousazadeh and I. Cohen "Short Utterances Diarization Using GMM Supervectors and Spectral Clustering" submitted to IEEE Transactions on Audio, Speech, and Language Processing.

Contents

List of Figures

List of Tables

Abstract

Speaker diarization is defined as the task of tagging different speakers within an unmarked speech sequence. Speaker diarizarion has attracted significant research effort in the last decade. Nevertheless, diarization of short utterances (2-5 seconds), particularly under noisy conditions, is still a very challenging task. In this dissertation, we introduce a speaker diarization system which is based on three components: A voice activity detection (VAD), utterances representation and spectral clustering.

Traditional VADs suffer from high false detection rates in noisy conditions. Therefore, we have implemented a unique VAD which aims to deal with noisy environments. It is based on extraction of special features, graph embedding and Laplacian pyramid representation.

State-of-the-art speaker diarization systems are usually based on statistical models. Their segmentation stage is obtained by a speaker change point detector that employs the Bayesian information criterion (BIC). It uses a penalized likelihood ratio test (LRT) to detect a speaker change point within a sliding window. Then, tagging different speakers is obtained by an agglomerative hierarchical clustering which also assumes a statistical model for each segment. In case of short utterances, the statistical models are not reliable and therefore state-of-the-art speaker diarization systems

are generally characterized by low performance in terms of speaker change point detection and confusion between speakers.

The proposed speaker diarization algorithm utilizes spectral clustering for final tagging. Each segment of speech is represented by a GMM mean supervector as well as by its first and second derivatives, which introduce additional aspect of discrimination between speakers. These supervectors are used as inputs to the spectral clustering technique. Spectral clustering exploits the eigenvectors of the similarity matrix of the data in order to perform dimensionality reduction. This technique enables to capture the most informative eigenvectors which improve the clustering performance and the robustness of the system to noise. Experimental results demonstrate a significant reduction of error rate in comparison with state-of-the-art and unsupervised diarization methods.

Notation

x	scalar
$\boldsymbol{x}$	column vector
$\boldsymbol{X}$	matrix
$x(n)$	time-domain signal
$(\cdot)^T$	transpose operation
$\|x\|$	absolute value
$\|\boldsymbol{x}\|$	Euclidian norm
$\|\boldsymbol{X}\|$	determinant
$\boldsymbol{X}^{-1}$	matrix inverse
∇	gradient
Δ	first Derivative
$\Delta\Delta$	second Derivative
$\mathbb{R}^m$	N-demential real-valued vector

Abbreviations

ACP	Average Cluster Purity
AIC	Akaike Information Criterion
AWGN	Additive White Gaussian Noise
BIC	Bayesian Information Criterion
DER	Diarization Error Rate
E-HMM	Evaluative Hidden Markov Model
EM	Expectation Maximization
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
HNR	harmonics to noise ratio
IMCRA	Improved Minima Controlled Recursive Averaging
KL	Kullback Leibler
LIA	Laboratiore Informatique d'Avignon
LLE	Locally Linear Embedding
LPP	Locality Preserving Projection
LRF	Likelihood Ratio Function
LRT	Likelihood Ratio Test
MAP	Maximum A-posteriori
MFCC	Mel Frequency Cepstral Coefficient
ML	Maximum Likelihood
MLE	Maximum Likelihood Estimation
NPE	Neighborhood Preserving Projections
PCA	Principal Component Analysis

PDF	Probability Density Function
SNR	Signal to Noise Ratio
STFT	Short Time Fourier Transform
UBM	Universal Background Model
VAD	Voice Activity Detection

Chapter 1

Introduction

1.1 Motivation and overview

Speaker diarization is defined as the task of tagging different speakers within an unmarked speech sequence and is also referred to as the "who spoke when?" problem. Speaker diarization has many applications, such as audio indexing and pre-processing module for speaker identification and verification systems [1–5]. In the last decade, the problem of speaker diarization has attracted a significant research effort. Nevertheless, diarization of short utterances and diarization in noisy environments are still considered as open issues that should be further researched.

Conventional speaker diarization systems include two main stages: Segmentation and clustering. The segmentation is usually obtained by BIC-based speaker change point detection. It is based on modeling of consecutive segments and employing likelihood ratio test (LRT) in order to decide on a speaker change point. This method has some drawbacks: First, models and LRT-based methods are very sensitive to noise. Second, using LRT requires a detection error to be empirically tuned and therefore has a trade-

off between pure segments and minimizing missing speaker change points. As consequence, applying BIC-based speaker change point detection on short utterances conversations leads to high miss detection rates. Since a fine segmentation is crucial for short utterances diarization, we suggest using voice activity detector (VAD) algorithm for this purpose.

In this thesis, we address the problem of short utterances diarization, particulary under noisy environments, which include stationary and transient noises. Transient interferences are short time interruptions such as keyboard typing, door knocking, sneezing, etc. Conventional VAD algorithms are usually assume statistical models for speech and noise signals, in order to apply some kind of likelihood test and decide whether the current frame contains speech or not. Moreover, it is assumed that noise is slowly varying with respect to speech. However, these assumptions are not reasonable when dealing with transients. Therefore, most of the conventional algorithms fail to detect speech in transient noisy environments and consider incorrectly transient noise as speech. In order to deal with transient interferences we developed a unique VAD which enables a fine segmentation, such that each segment contains only one active speaker. The next stage of the proposed system deals with the representation of each segment in order to achieve an appropriate discrimination between different speakers. We represent the segments by Gaussian mixture model (GMM) mean supervectors and their first and second derivatives. The supervectors are adapted from a universal background model (UBM) which aims to represent the general speaker. The adaptation causes shifts in local modes and regional centers of mass. Therefore, the adapted GMM

mean supervector actually represents local first-order differences between the UBM and the adapted GMM. In other words, the difference between the target GMM and the UBM holds a required information for discrimination between speakers. We also use the first and second derivatives of the GMM mean supervectors which further emphasize the difference between utterances of different speakers.

Conventional speaker diarization systems usually use agglomerative hierarchical clustering which assumes a statistical model for each segment. As discussed previously, depending only on statistical models is unreliable enough in cases of noisy environments and short utterances. Therefore, we suggest revealing the underlying structure of the data by using spectral clustering. The fact that natural data like speech can lie on a low-dimensional manifold [6] encourages the use of graph embedding or dimensionality reduction algorithms, which are very efficient in noisy conditions. It enables to capture the most informative dimensions and discards noise components. Therefore, it improves the robustness of the speaker diarization system to noise. In this work we examine a few methods for dimensionality reduction, but in the proposed speaker diarization system we choose the spectral clustering method. Spectral clustering is helpful for gaining insight into complex data sets which are described by high-dimensional features. It can also be used for estimating the number of involved speakers.

The contributions of our research are as follows. First, we develop a VAD algorithm with an improved performance in noisy environments compared to state-of-the-art algorithms. It also provides fine segmentation

which is necessary when dealing with rapid speaker change point conversations. Second, by using the supervectors and spectral clustering we improve the speaker diarization robustness to noise and enhance the discrimination between short utterances of different speakers.

1.2 Thesis Structure

The rest of the thesis is organized as follows: In Chapter 2, we survey recent algorithms for speaker diarization and provide a short theoretical background on graph embedding algorithms and spectral clustering. In Chapter 3, we present an algorithm for speaker diarizarion of short utterances under noisy environments. In Chapter 4, we present a novel algorithm for voice activity detection in presence of stationary and transient noise, which is based on Laplacian pyramid representation. In Chapter 5, we conclude our research and its contributions as well as discuss some future research directions.

Chapter 2

Related Work and Theoretical Background

2.1 Introduction

Although speaker diarization was widely researched in the last decade, short utterances diarization is still an open problem. Extracting reliable features or modeling successfully speech utterance are necessary in order to discriminate between speakers. It becomes harder, as the speech segment gets shorter. In addition, noisy environments which characterized by stationary and transient noises causes to a significant performance degradation. False detection of speech such that noise is considered wrongly as speech segments, may lead to unreliable model of the speaker. Therefore, rather then relying on some statistical model, revealing the underlying structure of the data can provide better understanding and eventually, better clustering performance, even in those challenging cases. It can be done by using a dimensionality reduction algorithm which is a wide researched field in signal processing. Its goal is to capture the most informative features of the data in a lower dimensional space.

This chapter is organized as follows: In Section 2.2 we review milestone and recent methods for speaker diarization. In Section 2.3 we present a state-of-the-art speaker diarization algorithm. In Section 2.4 we review the concept of dimensionality reduction algorithms and in particular, the spectral clustering which is a primary stage in our speaker diarization algorithm. Finally, in Section 2.5 we discuss the highlights of this chapter and elaborate the contribution of dimensionality reduction algorithms to speaker diarization systems.

2.2 Speaker Diarization

State-of-the-art speaker diarization methods are comprised of the same key subtasks: Speech detection, segmentation and clustering [4, 5, 7]. See an illustration in Fig. 2.1. These methods usually demonstrate acceptable

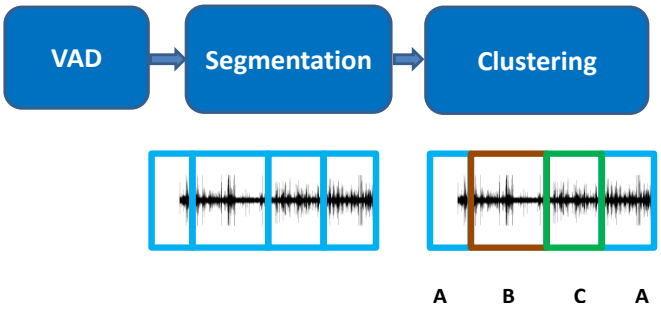


Figure 2.1: The baseline diarization system includes three main sub-tasks: Speech detection (VAD), segmentation and clustering. Each color demonstrates a specific speaker (A, B and C).

diarization performances [8, 9].

However, diarization of short utterances, particulary under noisy environments causes to significant performance degradation. The conventional approach for segmentation is applying a speaker change point detection

algorithm which is usually based on Bayesian information criterion (BIC). BIC-based speaker change point detector uses a penalized LRT to detect a speaker change point within a sliding window. The main drawback of this method is the requirement of detection error to be empirically tuned. It has a trade-off between creating pure segments and minimizing missing change points. Eventually, in case of rapid speaker changes conversation, it causes to high missed speaker change points rates. A detailed description of BIC-based speaker change point detector is introduced in Section 2.3.

While most diarization systems perform the segmentation and clustering separately, there are some methods which perform those stages jointly. The integrated approach inherently combines segmentation (speaker change point detection) and clustering phases, instead of using each of them separately. One implementation of this approach is the Laboratoire Informatique d'Avignon (LIA) diarization system. This system utilizes an evaluative hidden Markov modeling (E-HMM) of the conversation [10]. Each state of the HMM characterizes a different speaker, and the transitions indicate the changes between speakers. During the diarization, the HMM is generated using an iterative process which detects and adds a new state (i.e., a new speaker) at each iteration. The main advantage of such an integrated approach is that the system uses the whole audio sequence at each step [3]. Nevertheless, the difficulty in extracting reliable features and suitable statistical models in case of rapid speaker change points and noisy environments, is still relevant. Hence, this method also suffers from significant performance degradation when dealing with these challenging cases.

An important issue in the field of speaker recognition system (identification, verification, diarization, etc) is choosing the most suitable features in order to achieve appropriate speaker discrimination. Friedland et al. suggested to use prosodic and long-term features for speaker diarization [11]. Prosodic and long-term features refer to features that are extracted over regions longer than a frame (300ms and above). These features capture variations in intonation which are attributed to a specific speaker. One example for prosodic feature is the pitch frequency, which is the speaking fundamental frequency. It is influenced by the length and the mass of the vocal folds in the larynx [12]. The pitch frequency can discriminate successfully between male and female. However, in the same gender, the dynamic range is relatively narrow and different males or different females might have similar values of pitch frequencies. In addition to the pitch frequency, there are many other prosodic features; energy (changes in loudness), formants, harmonics-to-noise-ratio (HNR), etc. The prosodic features have some common drawbacks. First, calculation of prosodic features is relatively difficult and characterized by high computational complexity and load. Second, in case of rapid speaker change points (short utterances), those features became less effective. Third, there is still a significant performance degradation in case of noisy environments, in particular, transients which have some common properties of speech. To conclude, the contribution of prosodic features in the challenging cases of rapid speaker change points and noisy environments is lower.

Advanced methods in speech processing use a form of dimensionality reduction algorithms. The goal of this algorithms is to find predominant

features in a low-dimensional space, which capture the inherent structure of the data. These methods were proved to be accomplished in speaker diarization and clustering algorithms [13, 14]. Thanks to the assumption that speech can be well represented in a low dimensional space [6], combining dimensionality reduction algorithms in speech processing systems is a reasonable choice. A detailed explanation on dimensionality reduction is introduced in Section 2.4.

2.3 Baseline System

In this section we review the main stages of conventional speaker diarization systems. Most of the current systems utilize the same modules: Speech detection, speaker change point detection and speaker clustering (Fig. 2.1). However, they may differ in the way of modeling the data or because of additional intra modules, which aim to improve the system performance.

2.3.1 Speech Detection

The first stage in many speech and speaker processing algorithms is detecting the speech segments and discarding non-speech segments which usually contain noise. There is no valuable information in these frames, and at worst, they deteriorate the quality of extracted features and models. The speech detection is obtained by a VAD algorithm [15–21]. Practically, the VAD algorithm provides an index vector which determines whether a

given frame of time (20-30 ms) consists speech or not as follows:

$$\mathbb{1}_t = \begin{cases} 1 & \text{if frame } t \text{ contains speech signal;} \\ 0 & \text{if frame } t \text{ contains non-speech signal.} \end{cases}$$

2.3.2 Speaker Change Point Detection

The traditional approach for detecting speaker change point is the BIC-based detector. This technique detects a speaker change point within a sliding window using a penalized LRT to decide whether the data in a specific window is better modeled by a single distribution or by two different distributions. More specifically, let $H_{NoChange}$ and H_{Change} be the hypothe-

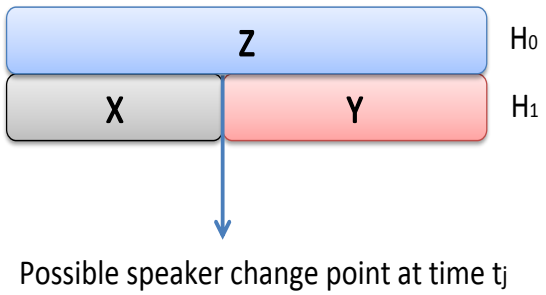


Figure 2.2: Schematic description of BIC-based speaker change point detector.

sis of speaker change point absence or presence at time t_j, respectively. Let L_0 and L_1 be the likelihoods of the observations given hypothesis $H_{NoChange}$ and H_{Change}, respectively, as follows:

$$L_0 = \sum_{i=1}^{N_x} \log P(\boldsymbol{x}_i|\theta_z) + \sum_{i=1}^{N_y} \log P(\boldsymbol{y}_i|\theta_z) \tag{2.1}$$

$$L_1 = \sum_{i=1}^{N_x} \log P(\boldsymbol{x}_i|\theta_x) + \sum_{i=1}^{N_y} \log P(\boldsymbol{y}_i|\theta_y) \tag{2.2}$$

when N_x and N_y are the total number of feature vectors of segments X and Y, respectively. θ_x, θ_y, θ_z are the models parameters of the probability density functions (PDF) which represent the segments X, Y and Z, respectively (See Fig. 2.2). The dissimilarity is estimated by:

$$S = L_1 - L_0 - P\frac{\lambda}{2}\log N_z \tag{2.3}$$

when λ is a penalty factor which tuned according to the data and P depends on the feature vectors dimension d as follows:

$$P = \frac{1}{2}(d + \frac{d}{2}(d+1)). \tag{2.4}$$

An existence of speaker change point at t_j is decided if $S > 0$ and vice versa. For further discussion see [22].

Due to the easy implementation, the BIC-based speaker change detector is used in many speaker diarization systems. However, this method has a significant performance degradation in cases of short speech utterances and noisy environments. In these cases, the estimated models are unreliable enough and there is a high risk of false detection or even missing the speaker change points location, such that the obtained segments can contain two or more different speakers, while practically they supposed to contain only one active speaker each. Besides, deep search implementation i.e., moving the window every few samples, is computationally expensive.

2.3.3 Agglomerative Hierarchical Speaker Clustering

The speaker change point detection algorithm provides a segmented conversation. The segments are used as input to the clustering algorithm. The

baseline system uses an agglomerative hierarchical clustering (bottom-up approach) which consists of the following steps:

1. Initializing leaf clusters of a tree with speech segments.
2. Computing pair-wise distances between each cluster by some criterion and merging the two closet clusters.
3. Updating distances of the remaining clusters.
4. Iterating stages 1 to 3 until a stopping criterion is met (usually, BIC-based stopping criterion).

An important issue to be addressed in the agglomerative hierarchical clustering method is how to determine which clusters are the closest. There are several methods for measuring the distance between two clusters, but the most popular is the generalized likelihood ratio (GLR) which is also used in this work as part of the baseline system implementation. For further discussion see [23].

2.4 Dimensionality Reduction

Many problems of information processing involve a form of dimensionality reduction algorithm. In general, these algorithms can be categorized as follows: Linear and nonlinear manifold embedding, Linear and nonlinear methods (algebraically), supervised and unsupervised, however the most crucial is the criterion of optimization. Some algorithms are based on constructing weight graph and adjacency matrix which characterize the distances between different data points in the graph [19, 24–26]. This analysis

enables learning and extracting the most important features of the data, which can represent the manifold in a low dimensional space. In other words, those algorithms are helpful for gaining insight into complex data sets described by high-dimensional features. An inherent property of many data sets is that a small set of natural parameters captures the important sources of variation in the data. Hence, extracting these features reveals the underlying structure and can improve exploration, visualization, modeling and clustering of the data points. In this section, we discuss briefly some recent and milestone methods of dimensionality reduction.

2.4.1 Principal Component Analysis Algorithm

Principal Component Analysis (PCA) algorithm is an eigenvector method designed to model linear variation in an high-dimensional space. It performs dimensionality reduction by projecting the original data onto the reduced dimensional linear subspace spanned by the leading eigenvectors of covariance matrix of the data. The objective of PCA is to find a set of orthogonal basis functions that holds the directions of maximum variance. It aims to preserve the global structure of the data and does not discover the underlying structure, the local structure. PCA is limited to a linear manifold, and thus, when the data lies near a linear manifold, PCA recovers a small set of coordinates which describes variations in the data aligned with this linear manifold.

Recently, advanced methods were developed under the assumption that the data lies near a nonlinear manifold, and thus represent the data as a weighted graph in order to better honor these nonlinearities: Isomap,

Locality linear embedding (LLE) and Laplacian eigenmap, just to name a few. The key intuition of these techniques is the importance of measuring distances along the manifold of the data and not through arbitrary dimensions in a high-dimensional space.

2.4.2 Locality Linear Embedding

Locality Linear Embedding (LLE) [27] is an unsupervised learning algorithm which represent high dimensional data point in low-dimensional space. LLE is an eigenvector method for nonlinear dimensionality reduction. It aims to underly nonlinear structure in high dimensional data by exploiting the local symmetries of linear reconstructions. In the following section, we briefly introduce this method.

Suppose that the data consists of n feature vectors; $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n$, each of dimensionality d, is sampled from a smooth underlying manifold. Assuming the data are sufficient, i.e., the manifold is well sampled, and each data point and its p nearest neighbors lie on a locally linear patch of the manifold. The local geometry of these patches are characterized by linear coefficients that reconstruct each data point from its neighbors. The reconstruction errors are measured by the cost function:

$$\mathcal{E}(\boldsymbol{W}) = \sum_i |\boldsymbol{x}_i - \sum_j W_{ij}\boldsymbol{x}_j|^2 \tag{2.5}$$

which sum-up the squared distances between data points and their reconstructions. The weights W_{ij} represent the contribution of the j-th data point to the i-th reconstruction. In order to compute the weights, the cost function is minimized subject to two constraints: Each data point can

be reconstructed only from its neighbors and the weight matrix rows are summed to one. The weights subject to these constrains are calculated by solving a least square problem. In the final stage of the algorithm, each high dimensional observation $\boldsymbol{x}_i$ is mapped to a low dimensional vector $\boldsymbol{y}_i$, representing global internal coordinates on the manifold. This is done by choosing D-dimensional coordinates $\boldsymbol{y}_i$ in order to minimize the embedding cost function:

$$\Phi(\boldsymbol{Y}) = \sum_i |\boldsymbol{y}_i - \sum_j W_{ij}\boldsymbol{y}_j|^2 \tag{2.6}$$

This cost function is also based on locally linear reconstruction errors as in (2.5), however the weights are fixed when optimizing the coordinates $\boldsymbol{y}_i$. This problem can be solved by sparse nxn eigenvector problem, whose bottom d non-zero eigenvectors provide an ordered set of orthogonal coordinated centered on the origin [28]. To conclude, the LLE consists of three main stages: Selecting neighbors of each data point, minimizing the cost function in (2.5) and calculating the weights, and finally mapping to the embedded coordinates. The LLE is sensitive to few parameters which are set in advance: The dimensionality to map to (D), the number of neighbours (p) and a regularization parameter, which is part of the weights calculation. If D is too high, the mapping enhances noise, and if it is too low, multiple data points will be mapped to the same low-dimensional vector. Setting high p leads to a behaviour which is similar to PCA because of loosing the nonlinear property, while setting small p will discard absolutely any global preserving key. Because of the assumption that close points in the high-dimensional space stay close in the low-dimensional space, the weights are preserved. In other words, the LLE holds a neighboring-preserving

mapping.

2.4.3 Spectral Clustering

Before diving into the spectral clustering algorithm, we briefly overview similarity graph notations.

Let $x_1, x_2, ..., x_n$ be a set of data points. The intuitive explanation of clustering is dividing the data points into several groups such that points in the same group are similar and points in different groups are dissimilar to each other, in terms of some metric. Conventional methods for data representation include constructions of a similarity graph $G = (V, E)$ when V is a set of vertices and E is a set of edges of the graph. Each vertex v_i in the graph represents a data point. Each edge e_{ij} between two vertices v_i and v_j carries a non-negative weight $w(i, j)$ which represents the similarity between the corresponding points. A similarity matrix $\boldsymbol{W}$ is a matrix whose (i, j)-th element equals to $w(i, j)$. We assume that the graph is undirected, i.e., $w(i, j) = w(j, i)$. In term of similarity graph, the goal is to find a partition of the graph such that the edges between different groups will have low weights and the edges within a group will have high weights.

The degree of a vertex $V_i \in V$ is defined as:

$$d_i = \sum_{i=1}^{n} w_{ij}. \tag{2.7}$$

The degree d_i holds the weight of a specific data point, x_i, within the set. For example, suppose that x_i is far from all other data points (i.e., an outlier), its degree is relatively low and so its influence on the adjacency matrix. This fact can improve the robustness of the algorithm to outliers.

The degree matrix $\boldsymbol{D}$ is defined as the diagonal matrix with the vertices degree $d_1, d_2, ..., d_n$ on the diagonal.

Spectral clustering is a class of techniques which utilize the eigen structure of a Laplacian matrix, in order to divide points into disjoint clusters. Data points in the same cluster have high similarity while points in different clusters have low similarity. Laplacian and weight matrices constitute an important issue in spectral clustering. The Laplacian matrix usually combines between the degree and the weight matrices, representing the intrinsic connections of the data points [29]. There are two types of Laplacian matrices: Un-normalized Laplacian and normalized Laplacian (symmetric or non-symmetric). Each type of Laplacian matrix satisfies different properties regard the connectivity between the data points. In the un-normalized method, this connection does not depend on the degree of vertex, i.e., on $d_i, i = 1, ..., n$. Therefore, in order to deal with data points which are broadly distributed, one should choose the normalized Laplacian rather than the un-normalized.

In this section, we introduce the spectral clustering algorithm according to Ng, Jordan, and Weiss [30]. This algorithm utilize the normalized and symmetric Laplcian matrix. The algorithm stages are described as follows:

1. Constructing the symmetric normalized Laplacian matrix $\boldsymbol{L}$ which is denoted here as $\boldsymbol{L}_{Spectral}$:

$$\boldsymbol{L}_{Spectral} = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2} \tag{2.8}$$

   where $\boldsymbol{W}$ is the weight matrix.

2. Computing the first K eigenvectors $\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_k$ of matrix $\boldsymbol{L}_{Spectral}$,

corresponding to the k largest eigenvalues. Let $\boldsymbol{U}$ be a matrix that consists of vectors $\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_k$ as its columns.

3. Constructing the matrix $\boldsymbol{Y}$ by normalizing the rows of matrix $\boldsymbol{U}$. Each row of $\boldsymbol{U}$ is re-normalized to have a unit length as follows

$$\boldsymbol{Y}_{ij} = \boldsymbol{U}_{ij} / (\sum_{j=1}^{n} \boldsymbol{U}_{ij}^2)^{1/2}. \tag{2.9}$$

4. Clustering the points $\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_n$ using the K-means algorithm, and obtain the clusters $C_1, C_2, ..., C_k$.

The spectral clustering includes two stages: Dimensionality reduction and clustering. The dimensionality reduction stage consists of calculating the eigenvectors and eigenvalues of the Laplacian matrix. It is a computationally efficient approach to nonlinear dimensionality reduction which holds locality preserving properties, leading to a natural clustering. It is worth to mention that in contrast to other dimensionality reduction algorithms, the spectral clustering has no specific constrains and is not limited to linear embedding or even linear patches.

2.4.4 Locality Preserving Projections Algorithm

Locality Preserving Projections (LPP) algorithm [31] is a linear dimensionality reduction algorithm which aims to preserve the local structure of a nonlinear manifold. Assuming that the high-dimensional data lies on a low dimensional manifold embedded in the ambient space, the LPP are obtained by finding the optimal linear approximations to the eigen-functions of the Laplacian-Beltrami operator on the manifold, building a graph in-

corporating neighborhood information of the data set. In this section we briefly introduce the LPP method.

Let $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n$ be n feature vectors and let G denotes a graph with n nodes, where node i represents the data point $\boldsymbol{x}_i$. The goal is representing $\boldsymbol{x}_i$'s in l-dimensional space. The LPP algorithm consists of the following steps:

1. Constructing undirected adjacency graph by setting an edge between nodes i and j if the vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are close, i.e., if $||\boldsymbol{x}_i - \boldsymbol{x}_j||^2 < \epsilon$.

2. Choosing the weights while using heat kernel. If nodes i and j are connected, the weight between them is calculated as:

$$\boldsymbol{W}_{ij} = exp(-||\boldsymbol{x}_i - \boldsymbol{x}_j||^2/t) \tag{2.10}$$

   where t is a real fixed number which controls how rapidly the weight matrix falls off with the distance between feature vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$.

3. Computation of Eigen-maps. Let $\boldsymbol{X} = \{\boldsymbol{x}_1\ \boldsymbol{x}_2\ ...\ \boldsymbol{x}_n\}$ be a matrix of features when $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n$ are the column feature vectors. Then, eigenvectors and eigenvalues for the generalized eigenvector problem:

$$\boldsymbol{X}\boldsymbol{L}_{LPP}\boldsymbol{X}^T\boldsymbol{a} = \lambda\boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{a} \tag{2.11}$$

   are calculated. When $\boldsymbol{D}$ is the degree matrix and $\boldsymbol{L}_{LPP} = \boldsymbol{D} - \boldsymbol{W}$ denotes the un-normalized Laplacian matrix. The solution of (2.11) is $\boldsymbol{a}_0, \boldsymbol{a}_1, ..., \boldsymbol{a}_{l-1}$ in a decreasing order of their eigenvalues. Finally, the reduced representation is $\boldsymbol{x}_i \rightarrow \boldsymbol{y}_i = \boldsymbol{A}\boldsymbol{x}_i$.

2.4.5 Neighborhood Preserving Embedding

Neighborhood preserving embedding (NPE) like LPP is a method to linearly approximate the eigenfunctions of the Laplace Beltrami operator. In fact, it can be considered as a linear approximation to the LLE algorithm [31], while LPP method is considered as a linear approximation to Laplacian eigenmaps method. The NPE algorithm procedures are similar to LPP method: Constructing the adjacency graph, computing the weight matrix and finally computing the projections. However, in the last step, the linear projections are calculated by solving the following generalized eigenvector problem:

$$\boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{a} = \lambda \boldsymbol{X}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{a} \tag{2.12}$$

where the matrix $\boldsymbol{M}$ defined as:

$$\boldsymbol{M} = (\boldsymbol{I} - \boldsymbol{W})^T(\boldsymbol{I} - \boldsymbol{W}). \tag{2.13}$$

Similar to LPP method, the solution of (2.12) is $\boldsymbol{a}_0, \boldsymbol{a}_1, ..., \boldsymbol{a}_{l-1}$ in a decreasing order of their eigenvalues. Finally, the reduced representation is $\boldsymbol{x}_i \rightarrow \boldsymbol{y}_i = \boldsymbol{A}\boldsymbol{x}_i$. NPE aims at preserving the local manifold structure, as well as at LPP method. However, LPP and NPE methods have different objective functions. LPP method aims to minimize the following objective function which is based on a neighboring-preserving criterion:

$$\sum_{i,j} (\boldsymbol{y}_i - \boldsymbol{y}_j)^2 w_{ij}. \tag{2.14}$$

As can be seen, a heavy penalty is paid if neighboring points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are mapped far apart.

NPE aims to minimize the objective function which is based on a reconstruction error:

$$\sum_i (\boldsymbol{y}_i - \sum_j w_{ij}\boldsymbol{y}_j)^2. \tag{2.15}$$

NPE assumes well sampled manifold such that the close area of each data point is well established with neighbors and the reconstruction from neighbors is efficient. However, in case of high distributed data points NPE is less reliable.

2.5 Discussion

In this chapter, we reviewed state-of-the-art speaker diarization algorithms. We concluded that current methods for speaker diarization fail to deal successfully with short utterances, and in particular under noisy environments. Then, we introduced the field of dimensionality reduction algorithms: PCA, LLE, NPE, LPP and spectral clustering. When comparing between the different methods, we can conclude the main insights as follows. First, All method except PCA are suitable for non-convex and nonlinear manifolds. In addition, they aim to preserve the local structure of the data in comparison to PCA which aims to preserve the global structure of the data. Second, the NPE is a linear approximation to LLE while LPP is a linear approximation to Laplacian eigenmaps method, which is part of spectral clustering (the dimensionality reduction stage). Third, the LLE and NPE are based on minimization of reconstruction error, assuming each data point is reconstructed by its neighbors, i.e., they take into

consideration also the globally structure of the nonlinear manifolds.

Assuming that speech can be represented in low dimensional space [6], using the method of spectral clustering or any other dimensionality reduction algorithm is a suitable choice. However, the spectral clustering also preserves the local structure of the data but does not hold any linearity or reconstructed constrains. In this thesis, we show that thanks to these characteristics, the spectral clustering is a suitable tool for short utterances diarization in noisy environments.

Chapter 3

Speaker Diarization Using Spectral Clustering

3.1 Introduction

In this chapter we introduce an unsupervised speaker diarization system, focusing on rapid speaker change points (short utterances) and noisy environments. Speaker diarization task can be divided into two sub-tasks: Segmentation and clustering. The goal of segmentation stage is to divide a speech sequence into utterances which contain one speaker each. Then, the clustering aims to merge segments of the same speaker. Conventional algorithms usually use BIC-based speaker change point detector [22] for segmentation purpose. The idea behind this method is using a penalized LRT for detecting a speaker change points within a sliding window. This method is very popular and demonstrates acceptable performance. However, in cases of short utterances and noisy environments, many speaker change points are missed. Mainely, due to the difficulty in extracting significant speaker features and models of short utterances.

For further processing of the obtained segments, most of the available

methods describe its acoustic feature vectors by a time-independent statistical model like a Gaussian model or a GMM. Recently, the concept of GMM mean supervector became very popular in speaker verification and identification algorithms [32]. Therefore, combining this method in speaker diarization systems is a reasonable choice. We propose to represent each segment of speech by a supervector which consists of GMM mean supervector and its first and second derivatives. The derivatives hold a significant information which can help us to distinguish between speakers.

Thanks to the fact that speech concentrates on low-dimensional structure [6], we chose to apply a dimensionality reduction algorithm which captures the main features of the data, discarding outliers and noise. A promised method of graph embedding algorithm is the spectral clustering [30]. It introduces clustering as a graph partitioning problem without assumption on the clusters or on the manifold. Spectral clustering main advantages are: First, the clusters analysis relies on explicating the eigenstructure of a similarity matrix, rather than estimating some explicit model of data distribution. It covers the inherent structure of the data without assuming any type of model in advance. Second, The spectral clustering is characterized by convenience algebra, and last the number of clusters can be estimated from the spectrum of eigenvalues.

In this chapter we present a new method for speaker diarization system which focuses on the challenging cases of short utterances and noisy environments. The chapter is organized as follows: In Section 3.2 we formulate the problem of speaker diarization, and the proposed algorithm is described in Section 3.3. Experimental results and performance evaluation

of the proposed speaker diarization system are presented in Section 3.4. Finally, we conclude this chapter in 3.5

3.2 Problem Formulation

Let the signal $y(n)$ be a recorded conversation, measured by a microphone. We assume that $y(n)$ is comprised of clean speech, additive stationary and additive transient noise signals as follows:

$$y(n) = x_{sp}(n) + x_{st}(n) + x_{tr}(n) \tag{3.1}$$

when $x_{sp}(n)$, $x_{st}(n)$ and $x_{st}(n)$ are clean speech, additive stationary noise and transient noise, respectively. The outputs of the speaker diarization system are segmented and indexed conversations, when each segment carries a specific index that attributes to the relevant speaker.

In our research, we focus on short utterances diarization in noisy environments. Therefore, we assume challenging test signals which are characterized by rapid speaker change points and noisy environments include stationary and transient noises. We also assume non-overlapping speech segments and relatively short conversations (up to 15 minutes).

3.3 Proposed Algorithm

The proposed algorithm consists of three main components, where each component plays a significant role in the diarization system. First, we apply fine segmentation using a unique VAD algorithm which is described in Chapter 4. Then, we represent each utterance by a supervector, and last we employ spectral clustering technique for final tagging. In this section, we

describe in detail each component of our method. A schematic description of the proposed diarization system is presented in Fig. 3.1.

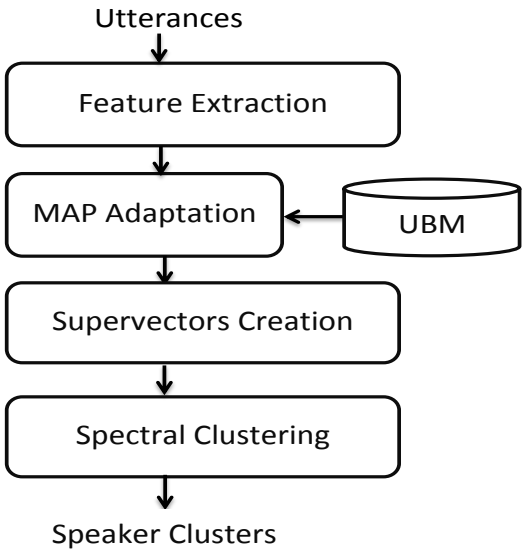


Figure 3.1: A schematic description of the proposed method. Extracting the feature vectors, adapting a GMM supervectors and applying spectral clustering.

3.3.1 Segmentation

Generally, segmentation is the first stage in all speaker diarization algorithms. It divides the speech sequence into segments which consist of one active speaker each. Most of the algorithms utilize the BIC-based speaker change point detector which suffers from significant performance degradation in cases of rapid speaker change points and noisy environments. Other algorithms support on equal length segmentation, which divide the speech sequence into equal segments of few seconds (1-3 sec). However, in order to deal with short utterances, it is necessary to provide fine segmentation that can introduce isolated utterances, consisting of one active speaker each. Therefore, we propose to utilize a VAD algorithm for speech

segmentation purpose.

The VAD algorithm provides a vector of indices which determine whether a specific frame contains speech or not. Most of conventional and state-of-the-art VAD algorithms have significant performance degradation in presence of transients and low signal to noise ratio (SNR) level of stationary noises. The performance degradation is characterized by high false rates, i.e., non-speech frames (noise frames) detected as speech frames. When pure noise segments are incorrectly detected as part of the speech segment, the obtained model is damaged and represents the speaker badly. As consequence, the ability to discriminate between speakers is degraded.

Hence, we implemented a unique VAD algorithm which handles with very noisy environments. Indeed, using VAD algorithm enables fine segmentation which is necessary in order to deal with short utterances, however there are cases in which the VAD algorithm divides one utterance into extremely short segments (around 300ms long). In most of the cases, these segments are a result of a brief break during the pronunciation of a word. Hence, we postulate that these segments are part of the current speaker. It is a reasonable assumption because speaking a word does not last less than half a second and a syllable or any other type of noise, like coughing, is not one of the speaker diarization interests. Capturing speech-only segments and relevant characteristics from each utterance is the key of distinguishing between different speakers.

3.3.2 Feature Extraction

Choosing the relevant features is one of the key stages in any data analysis problem and particulary in speaker diarization task. The ability to discriminate between different speakers highly depends on the chosen features. Kinnunen et al. [33] divide the features space into three general categories: High-level features, prosodic and spectro-temporal features and last short-term spectral and voice source features, see Fig. 3.2. High-level features attempt to capture conversation-level characteristics of speakers, such as characteristic use of words. Using these features while dealing with speaker verification problem is relatively easy, since high amount of training speech data from each speaker is available. However, in speaker diarization problem and particulary in short utterances diarizarion, it is almost impossible to rely on such kind of features. Pitch frequency is a popular prosodic feature. In general, the pitch frequency can be very useful as a discriminator between speakers of different gender. However, it turns into less reliable when mixed-gender population is involved in the conversation. In this case, the dynamic range of pitch values is relatively narrow and discrimination between speakers becomes difficult. In addition, the pitch frequency estimator is very sensitive to noise. In some variations, the spectro-temporal features hold an important information like changes of spectral features, which are also used in the proposed algorithm. To conclude, the short-term spectral features seem to be the most suitable choice. However, the significant drawback in short-term spectral features is the performance degradation in noisy environments.

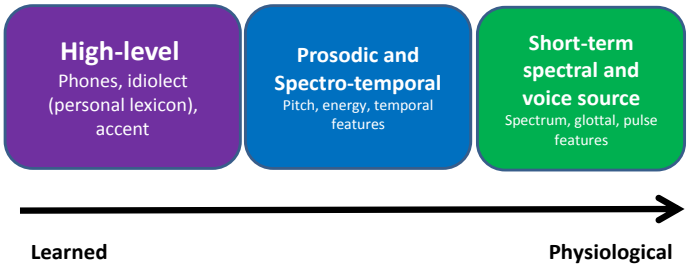


Figure 3.2: Partition of features; from learned to physiological features.

In the proposed speaker diarization algorithm, we utilize the Mel frequency cepstral coefficients (MFCCs). The MFCCs are a result of a cosine transform of the real logarithm of the short term energy spectrum expressed on a mel-frequency scale. It is based on the human auditory system and therefore found to be a successfully method for speech and speaker recognition systems [4, 5, 34, 35].

The noisy speech signal $y(n)$ as described in (3.1) is divided into time frames of 32ms long with overlap of 16ms, and the MFCCs are calculated for each frame. Let $\mathbf{Y}_m(k,t)$ $(t = 1, ..., N; k = 1, ..., K_m)$ be the absolute value of MFCCs where K_m is the number of frequency bins and N is the number of frames. As can be deduced, MFCCs describe only the power spectral envelope of a single frame. However, speech would also have information in the dynamics i.e., the trajectories of the MFCCs over time. Hence, we add the first and second derivatives of the MFCCs to our feature vector. Therefore, each frame is represented by a $3K_m$-dimension

column vector $\boldsymbol{Y}(:,t)$ as follows:

$$\boldsymbol{Y}(:,t) = \begin{bmatrix} \boldsymbol{Y}_m(:,t) \\ \Delta\boldsymbol{Y}_m(:,t) \\ \Delta\Delta\boldsymbol{Y}_m(:,t) \end{bmatrix} \tag{3.2}$$

where t is the frame index, and $\Delta\boldsymbol{Y}_m(:,t)$ and $\Delta\Delta\boldsymbol{Y}_m(:,t)$ are the first and second derivatives of the MFCCs, respectively. We compute the MFCCs in 19 frequency bands on the Mel scale. Hence, each frame is represented by a 57-dimension column vector.

3.3.3 Utterances Representation

Any utterance is characterized by a set of feature vectors as described in Sub-section 3.3.2, while the number of feature vectors depends on the utterance length. The objective of the utterance representation stage is to represent each utterance by using its acoustic feature, such that the discrimination between different utterances of different speakers will be possible. Most of the available methods describe the acoustic feature vectors by a time-independent statistical model like a Gaussian model or a GMM. Due to its uni-modal nature, a single Gaussian is far from being sufficient for modeling the utterance acoustic features probability distribution. Thus, a GMM is preferred. Theoretically, GMM can approximate any continuous probability density function (PDF) arbitrarily close, given a sufficient number of Gaussian components.

In the proposed algorithm, the representation of an unlabeled utterance is based on GMM mean supervector method. The GMM mean supervector is a long feature vector which consists of stacked mean components of the

GMM. It can be considered as a transformation of an utterance to a high dimensional vector. The process of creating the GMM mean supervectors is introduced in Fig. 3.3 and in the following sections.

Each utterance is modeled by a M-component GMM with the PDF

$$Pr(X|\theta) = \sum_{i=1}^{M} \omega_i \mathcal{N}(X|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \tag{3.3}$$

where $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$ consists of N acoustic feature vectors of dimension d, M is the number of components, ω_i is the i-th mixture weight, and $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ is a multivariate Gaussian PDF. The i-th mixture weight ω_i can be interpreted as *a-priori* probability that a given observation comes from the source governed by the i-th Gaussian distribution. The parameters $\theta = \left\{\omega_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right\}$, is estimated using the expectation-maximization (EM) algorithm [36] based on the maximum likelihood estimation (MLE).

The number of extracted acoustic feature vectors from a short utterance is quite small for training a single GMM. Hence, in order to successfully avoid over fitting, we utilize a pre-trained GMM known as universal background model (UBM), and adapt it to the acoustic features of a given utterance. Usually, the UBM is trained on hours of speech and pretends to represent a general speaker. Training on different speakers during sufficient hours of speech makes the UBM more reliable in representing a general speaker. However, in our system, we suggest to train the UBM on tested audio sequences. Training the UBM on the target conversation has some advantages: First, it enables to model the UBM by less Gaussian components than the common UBM. Second, the MAP adaptation requires significantly less iterations, and last the target UBM reflects a

suitable model which also takes into account the noisy environments.

The adaptation process is implemented by maximum a-posteriori (MAP) adaptation algorithm, which is a well-known method. The idea is to start with *a-priori* model (the UBM) and then apply estimation process, similar to the EM algorithm. The MAP adaptation is also very popular in various speaker recognition problems.

Let $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$ be feature vectors extracted of a given utterance. In the first step, the implementation stages are described as follows: Computing the posterior probability of each unlabeled vector, falling into every Gaussian component. More specifically, for any mixture $i = 1, 2, ..., M$ of the UBM, the following probabilities are computed:

$$Pr(i|\boldsymbol{x}_t) = \frac{\omega_i \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\sum_{k=1}^{M} \omega_k \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}. \tag{3.4}$$

Then, using $Pr(i|\boldsymbol{x}_t)$ and $\boldsymbol{x}_t$ in order to compute the sufficient statistics for the weight, mean and variance parameters of each mixture as follows:

$$n_i = \sum_{t=1}^{N} Pr(i|\boldsymbol{x}_t) \tag{3.5}$$

$$\boldsymbol{E}_i = \frac{1}{n_i} \sum_{t=1}^{N} Pr(i|\boldsymbol{x}_t) \boldsymbol{x}_t \tag{3.6}$$

$$\boldsymbol{E}_i^{\,2} = \frac{1}{n_i} \sum_{t=1}^{N} Pr(i|\boldsymbol{x}_t)\, diag(\boldsymbol{x}_t \boldsymbol{x}_t') \tag{3.7}$$

where n_i, $\boldsymbol{E}_i$ and $\boldsymbol{E}_i^{\,2}$ are the sufficient statistics for the weight, mean and variance, respectively. It is worth to mention that we assume diagonal covariance matrices.

In the second step, the sufficient statistics which were calculated previously (3.5)-(3.7) are combined with the prior model, and the adapted

parameters for mixture i are created as follows [37]:

$$\hat{\omega}_i = [\alpha_i n_i/N + (1-\alpha_i)\omega_i]\delta \tag{3.8}$$

$$\hat{\boldsymbol{\mu}}_i = \beta_i \boldsymbol{E}_i + (1-\beta_i)\boldsymbol{\mu}_i \tag{3.9}$$

$$\hat{\boldsymbol{\sigma}}_i^2 = \gamma_i \boldsymbol{E}_i{}^2 + (1-\gamma_i)(\boldsymbol{\sigma}_i^2 + \boldsymbol{\mu}_i^2) - \hat{\boldsymbol{\mu}}_i^2. \tag{3.10}$$

The adaptation coefficients $\alpha_i, \beta_i, \gamma_i$ control the balance between the old and new model. They are data dependent and are determined for each Gaussian component and model parameter as:

$$\begin{aligned} \alpha_i &= n_i/(n_i + r^\alpha) \\ \beta_i &= n_i/(n_i + r^\beta) \\ \gamma_i &= n_i/(n_i + r^\gamma) \end{aligned} \tag{3.11}$$

where $r^\alpha, r^\beta, r^\gamma$ are fixed relevance factors. In our system we adapt only the means, therefore we set the relevance factors r^α and r^γ to zero. The scale factor δ is computed over all new mixture weights in order to guarantee that the weights sum up to unity. To conclude, the new sufficient statistics are used for updating the old UBM sufficient statistics for each mixture component. Then, the adapted parameters are computed corresponding to the new utterance.

The MAP adaptation causes shifts in local modes and regional centers of mass. Hence, the adapted GMM mean supervector actually represents local first-order differences between the UBM and the adapted GMM. This insight implies on the speaker separation ability which is attributed to the GMM mean supervector. The feature vectors which are extracted from short utterances have limited differences such that the MAP adaptation

provides quite similar GMMs. Therefore, using a target UBM which is based on the given conversation can emphasize the variety between GMMs, which is necessary in order to discriminate between speakers.

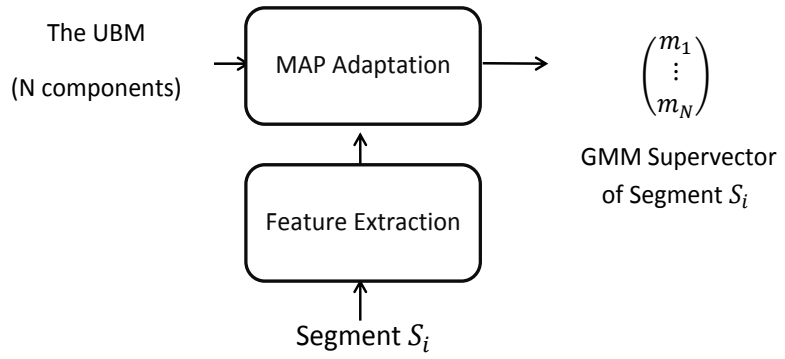
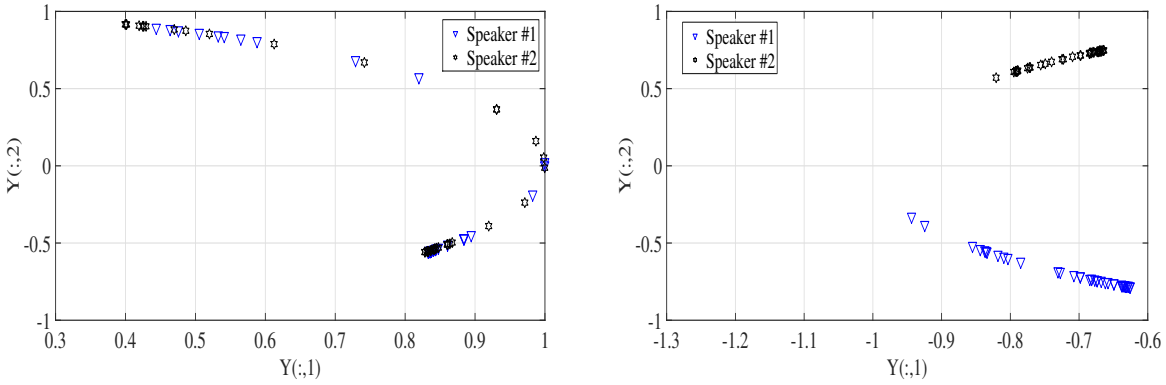


Figure 3.3: The principal of GMM mean supervector creation: Extracting feature vectors from an input utterance, applying MAP adaptation using UBM, and finally concatenating the means component. In the diagram, the UBM consists of N Gaussian (i.e., N mean vectors).

The GMM supervector is obtained by a concatenation of the adapted GMM mean components. It is used in many speech processing applications. For example, Campbell et al. reported successful verification performance compared to other methods [32]. For enhancing the different between GMM mean supervectors, we propose to utilize also the first and the second derivatives of the GMM mean supervectors. The trajectories of these supervectors over time emphasize the difference between consecutive utterances and improves the discrimination between speakers. Hence, every utterance is modeled by the vector:

$$\boldsymbol{SV}(:,a) = \begin{bmatrix} \boldsymbol{SV}_m(:,a) \\ \Delta\boldsymbol{SV}_m(:,a) \\ \Delta\Delta\boldsymbol{SV}_m(:,a) \end{bmatrix} \tag{3.12}$$

where a is the utterance index, $\boldsymbol{SV}_m(:,a)$ is the GMM mean supervector, and $\Delta\boldsymbol{SV}_m(:,a)$ and $\Delta\Delta\boldsymbol{SV}_m(:,a)$ are the first and second derivatives of the GMM mean supervectors, respectively. Fig. 3.4 demonstrates the contribution of the proposed representation method. Thanks to the usage of the derivatives, an improved separation between the data points and better clustering are achieved.



(a) **(b)**

Figure 3.4: Scatter plots in the resulted spectral clustering space. The tested conversation composed of 160 utterances (data points) under a stationary noisy environment at SNR level of 5dB. The axis $\boldsymbol{Y}(:,1)$ and $\boldsymbol{Y}(:,2)$ are the first and second normalized eigenvectors of the Laplacian matrix, respectively, which are corresponding to the first largest eigenvalues. (a): Scatter plot of GMM mean supervectors. (b): Scatter plot of proposed supervector method which combines the first and second derivatives of the GMM mean supervector.

3.3.4 Clustering

In the proposed algorithm, we utilize spectral clustering. The first step in spectral clustering is representing the data by a similarity graph. Let $G = (V, E)$ be a weighted graph where V is the set of the vertices and E is the set of the edges of the graph. Each vertex v_i in the graph represents a data point. Each edge e_{ij} between two vertices v_i and v_j carries a non-negative

weight $\boldsymbol{W}(i,j)$ which represents the similarity between the corresponding points. A similarity matrix $\boldsymbol{W}$ is a matrix whose (i,j)-th element equals to $\boldsymbol{W}(i,j)$. We assume that the graph is undirected, i.e., $\boldsymbol{W}(i,j) = \boldsymbol{W}(j,i)$. Ng et al. [30] proposed a spectral clustering method using the eigenvectors of the normalized Laplacian matrix:

$$\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}. \tag{3.13}$$

Matrix $\boldsymbol{D}$ is a diagonal matrix whose i-th diagonal element equals to $\sum_{j=1}^{N} \boldsymbol{W}(i,j)$ (i.e., $\boldsymbol{D} = \mathrm{diag}\,(\boldsymbol{W}\mathbf{1})$ where $\mathbf{1}$ is a column vector of ones). More specifically, let K be the number of clusters and $\boldsymbol{U}$ be a matrix consists of the first K eigenvectors of $\boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$ corresponding to K largest eigenvalues of $\boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$. The clustering is obtained by applying the weighted k-means algorithm on matrix $\boldsymbol{Y}$ rows which is a normalization of matrix $\boldsymbol{U}$ (for further information see 2.4.3). An important issue which must be addressed is the similarity matrix definition. Let $\boldsymbol{SV}(:,i)$ and $\boldsymbol{SV}(:,j)$ be supervectors which are extracted from the i-th and the j-th utterances, respectively. Let the term d_{ij} defines a distance metric. The proposed similarity matrix is based on the Gaussian kernel and is calculated as follows:

$$\boldsymbol{W}_{ij} = \exp(-d_{ij}^2/\sigma^2) \tag{3.14}$$

where the scaling parameter σ is a fixed parameter which controls how rapidly the similarity matrix $\boldsymbol{W}_{ij}$ falls off with the distance between the supervectors: $\boldsymbol{SV}(:,i)$ and $\boldsymbol{SV}(:,j)$. Usually, the scaling factor is manually set as a fixed value. Ng et al. [30] suggested picking the scaling factor automatically by searching the value which obtains the smallest distortion

after clustering $\boldsymbol{Y}$'s rows. Zelnik-Manor et al. [38] suggested to calculate a local scaling parameter for each data point, instead of a single scaling parameter. However, both of these methods require high computational complexity and memory. In the proposed algorithm, we choose the scaling factor to be the mean of all calculated distances:

$$\sigma = \underset{i,j}{mean} \left(d_{ij} \right). \tag{3.15}$$

As mentioned previously, the term d_{ij} in (3.14) is determined as a distance metric between different data points (utterances). In the proposed algorithm we compare between two types of metric distance methods: Likelihood-based and vector-based methods. The chosen likelihood-based distance metric is the KL divergence which is a natural measurement between probability distributions and is successfully used in speaker clustering applications [39]. The chosen vector-based distance metric is the cosine distance which is also very popular in speaker verification [40, 41] and clustering tasks [13].

3.3.5 Likelihood-Based Distance Metric (KL Divergence)

Suppose we have two utterances Utt_a and Utt_b with their corresponding adapted GMMs GMM_a and GMM_b, respectively. The GMMs have M components each. In addition, let $\boldsymbol{SV}(:, a)$ and $\boldsymbol{SV}(:, b)$ be the GMM mean supervector of Utt_a and Utt_b, respectively. The KL divergence measures the distance between probability distributions and is defined as:

$$KL(GMM_a, GMM_b) = \int_{\Re^n} GMM_a(x) \log \left(\frac{GMM_a(x)}{GMM_b(x)} \right) dx. \tag{3.16}$$

The divergence can be bound using log-sum inequality [42] as follows:

$$KL(GMM_a, GMM_b) \leq \sum_{i=1}^{M} \omega_i KL(\mathcal{N}(\cdot; \boldsymbol{\mu}_i^a, \boldsymbol{\Sigma}_i^a), \mathcal{N}(\cdot; \boldsymbol{\mu}_i^b, \boldsymbol{\Sigma}_i^b)) \tag{3.17}$$

where $\boldsymbol{\mu}_i^a$ and $\boldsymbol{\mu}_i^b$ are the i-th Gaussian component mean vectors of GMM_a and GMM_b, respectively. Assuming diagonal covariances, the approximation in (3.17) can be calculated by a closed form as:

$$\sum_{i=1}^{M} \omega_i KL(\mathcal{N}(\cdot; \boldsymbol{\mu}_i^a, \boldsymbol{\Sigma}_i^a), \mathcal{N}(\cdot; \boldsymbol{\mu}_i^b, \boldsymbol{\Sigma}_i^b)) = \frac{1}{2} \sum_{i=1}^{M} \omega_i (\boldsymbol{\mu}_i^a - \boldsymbol{\mu}_i^b)^T \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{\mu}_i^a - \boldsymbol{\mu}_i^b). \tag{3.18}$$

The last inequality is then:

$$0 \leq KL(GMM_a, GMM_b) \leq d_{KL}(\boldsymbol{SV}(:, a), \boldsymbol{SV}(:, b)) \tag{3.19}$$

where

$$d_{KL}(\boldsymbol{SV}(:, a), \boldsymbol{SV}(:, b)) = \frac{1}{2} \sum_{i=1}^{N} \omega_i (\boldsymbol{\mu}_i^a - \boldsymbol{\mu}_i^b)^T \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{\mu}_i^a - \boldsymbol{\mu}_i^b). \tag{3.20}$$

It can be deduced that if the distance between $\boldsymbol{SV}(:, a)$ and $\boldsymbol{SV}(:, b)$ is small, so is the corresponding divergence. Unlike the original term of KL divergence, the obtained term in (3.18) is symmetric and suitable for representing an undirected graph.

3.3.6 Vector-Based Distance Metric (Cosine Metric)

Given two supervectors $\boldsymbol{SV}(:, a)$ and $\boldsymbol{SV}(:, b)$, the cosine metric is defined by:

$$d_{cosine}(\boldsymbol{SV}(:, i), \boldsymbol{SV}(:, j)) = 1 - \frac{\boldsymbol{SV}(:, i)^T \boldsymbol{SV}(:, j)}{\sqrt{\boldsymbol{SV}(:, i)^T \boldsymbol{SV}(:, i)} \sqrt{\boldsymbol{SV}(:, j)^T \boldsymbol{SV}(:, j)}}. \tag{3.21}$$

The cosine metric considers only the angle between the two supervectors and neglects their magnitudes. Hence, due to the fact that speaker information is not part of the magnitude, the cosine metric is a suitable choice. Besides, channel properties might affect the magnitude of GMM supervectors, therefore the use of cosine metric can also improve the robustness of the algorithm to noise. In addition, the cosine metric enables to use the proposed supervector, while the KL divergence requires PDF (i.e., GMM mean supervector).

3.3.7 Estimating Number of Speakers

In the proposed algorithm, we assume that number of speakers, K, is unknown. The analysis introduced by Y. Ng et al. shows that the first eigenvalues, with the highest magnitude, of the normalized Laplacian matrix i.e., $\lambda_K, \lambda_{K-1}..., \lambda_1$, will be a repeated eigenvalue of magnitude 1 [30]. In addition, it is shown that those eigenvalues are corresponding to the number of clusters. However, in case of unclear separated groups, the magnitudes are deviated. Basing on perturbation theory, there is still a gap between the $(K+1)$-th and the K-th eigenvalues. Hence, the highest gap between consecutive eigenvalues implies on the number of clusters [43]. We set the number of speakers as the number K which provides the maximum value of $\lambda_{(K)} - \lambda_{(K+1)}$. Fig. 3.5 demonstrates the discussion above. For example, in case of six-speaker conversation, the gap between the 7-th and the 6-th eigenvalue is the highest (above 0.55). In case of ideal separated clusters, all six first eigenvalues should have value of 1, but in non-ideal case there is a slight decay from the third eigenvalue, such that the sixth eigenvalue

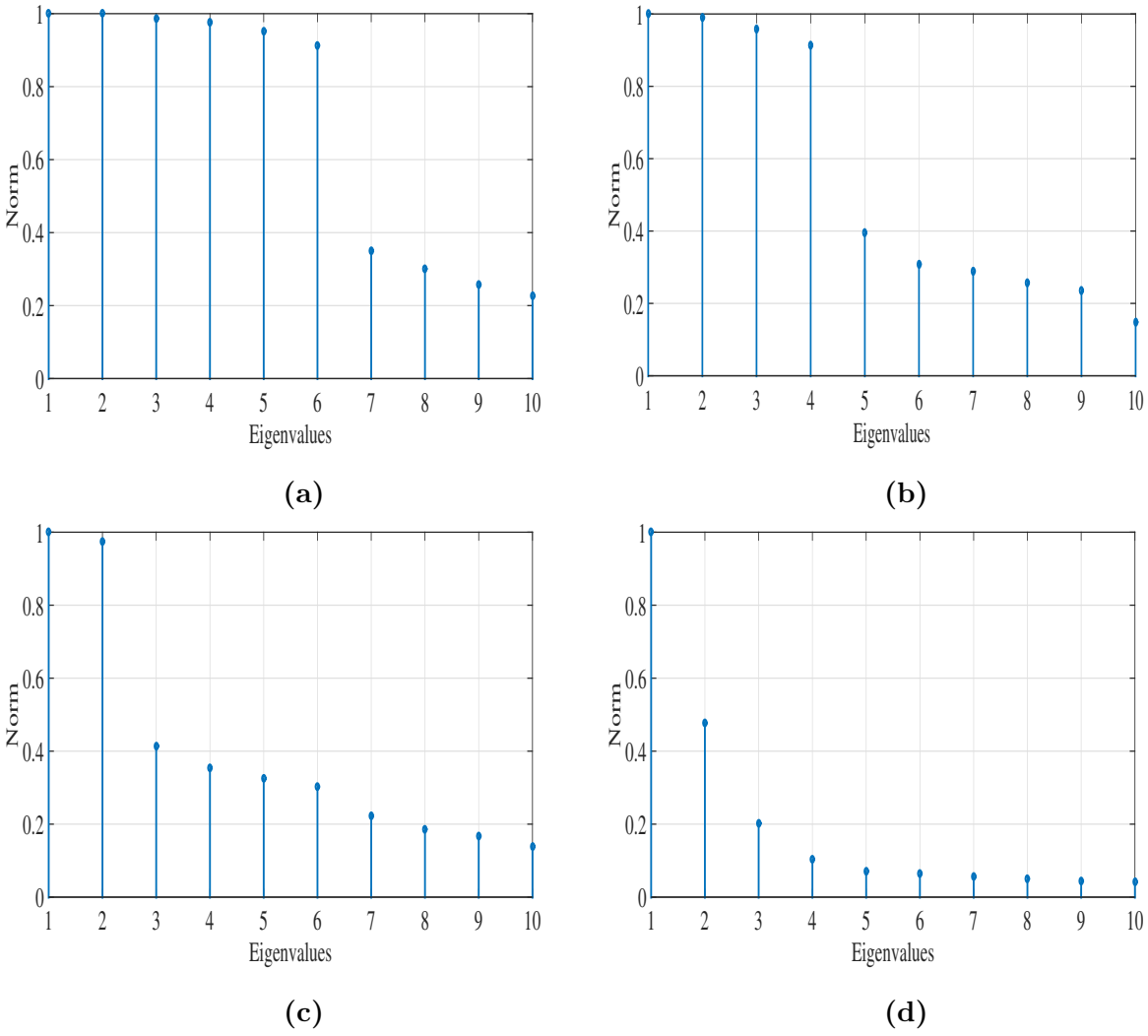


Figure 3.5: The eigenvalue spectrum of the normalized Laplacian matrix. When the groups number is k, the gap between the k-th eigenvalue and the $(k+1)$-th eigenvalue is higher then any other gap. (a): An example of six-speaker conversation. (b): An example of four-speaker conversation. (c): An example of two-speaker conversation. (d): An example one-speaker conversation

is equal to 0.91.

A different method for estimating the number of speakers is based on penalized BIC [44]. First, the algorithm is initialized by setting the number of speakers between 1 to 20 speakers. Then, the number of speakers N_{sp} is selected such that the following term is maximized:

$$BIC(Q) = \log L(X|Q) - s\frac{m}{2}N_{sp}\log N_x. \tag{3.22}$$

When Q is a model composed of N_{sp} models, N_x is the total number of speech frames, s is a tuning parameter which empirically chosen, and m is a parameter that depends on the complexity of the speaker models. An interesting point which should be addressed here is the case of one-speaker conversation. In case of one active speaker, the spectral clustering recognizes that only one cluster is relevant (see Fig. 3.5d). Although one speaker conversation is not the main purpose of speaker diarization systems, a conventional method may fail in such a sanity check.

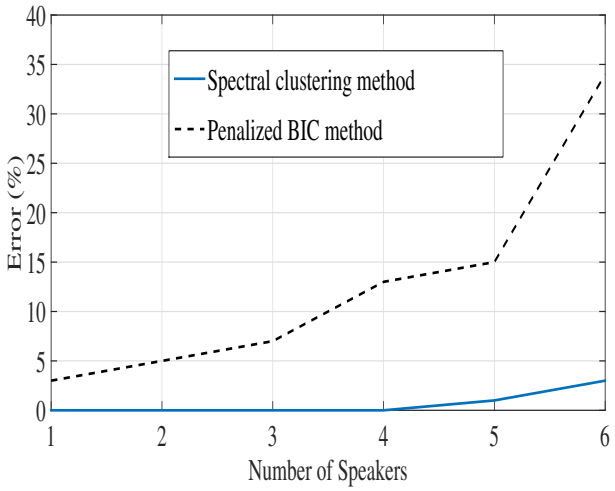


Figure 3.6: Estimating number of speakers using spectral clustering method and penalized BIC-based method. It is clearly seen that the estimation error is significantly lower when relying on the inherent structure of the data and not on models only.

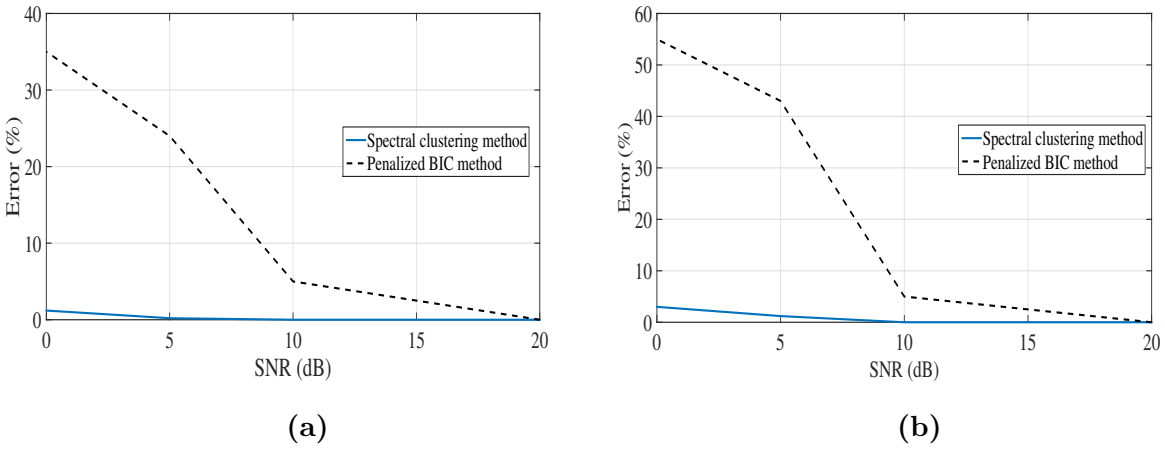


Figure 3.7: Estimation error rates under noisy environments using spectral clustering approach and penalized BIC-based method. (a): Four-speaker conversation. (b): Six-speaker conversation.

A comparison between the spectral clustering approach and the penalized BIC approach is concluded in Fig. 3.6. The Spectral clustering approach relies on analyzing the eigen-structure of an affinity matrix, while the penalized BIC method relies on estimating an explicit model on data distribution, which is degraded significantly in presence of noise. Therefore, using the spectrum of eigenvalues introduces better results. Moreover, the BIC-based method introduces a significant performance degradation as the number of speakers involved in conversation increases, and particulary under noisy environment, as can be seen in Fig. 3.7. Short utterances and noisy environments makes model-based methods to be less reliable. In conclusion, the inherent structure of the speech utterances enables to analyze the data better than any model-based method; estimating number of speakers and obtaining quality separation and clustering.

3.4 Experimental Results

3.4.1 Experimental Setup

We have examined the proposed algorithm on synthetic and recorded signals. In order to create relevant synthetic audio sequences, we used concatenated segments based on TIMIT [45] and FESTVOX [46] databases. The sequences are built such that speaker changes occur every 3.3 seconds, averagely. The maximum time between speaker change points is approximately 5 seconds. At each simulation, we examined our system on 500 synthetic speech sequences, including all possible gender combinations. The synthetic speech sequences are comprised of 2-6 speakers (100 sequences for each configuration). While using synthetic signals, we dealt with artificial boundaries by smoothing the transitions employing high-pass filter. Furthermore, in order to increase the diarization difficulty, we concatenated different males or females with a similar accent, while using the TIMIT database. In a separate experiment, we added different stationary noises including additive white Gaussian noise (AWGN) at several levels of SNR: 0dB, 5dB, 10dB, and 20dB, and babble noise. We also added transient noises such as door knock, metronome and keyboard stroke, which are taken from FREESOUND1 database [47]. All speech signals were sampled at 16 kHz and normalized to unity as their maximum. Likewise, we have also examined our proposed system on real data, we have recorded 50 conversations involving 2-4 speakers in a natural environment (car, living room, shopping mall, restaurants and cell phone).

We evaluate the performance of each method using two measures: Di-

arization error rate (DER) and average clustering purity (ACP). Both of them are well-known measurements for evaluation of speaker diarization systems.

The DER is the ratio of incorrectly detected speaker time to total speaker time and is given as the time-weighted sum of the following three error types:

1. Miss (M) - classifying speech as non-speech.

2. False Alarm (FA) - classifying non-speech as speech.

3. Confusion (C) - confusing one speaker with another.

Let T_M, T_{FA}, T_C, be the time length of missing speech intervals, false alarm intervals and wrongly detected speaker intervals, respectively. T_{Ref} is the total length of all speech segments in the ground true. The DER is calculated as [48]:

$$DER = \frac{T_{FA} + T_M + T_C}{T_{Ref}}. \tag{3.23}$$

It can be deduced that the clustering error is related only to T_C while T_M and T_{FA} evaluate the performance of the VAD algorithm. Hence, clustering which is a main stage in speaker diarization systems has to be evaluated separately. Therefore, we use the average cluster purity measurement. The ACP is introduced in [49] and determined as:

$$\begin{aligned} ACP &= \frac{1}{N}\sum_{i=1}^{S} p_i n_i \\ p_i &= \sum_{j=1}^{R}\left(\frac{n_{ij}}{n_i}\right)^2 \end{aligned}$$

where p_i is defined as the cluster purity of cluster i, R is the number of involved speakers in the conversation, S is the number of clusters, n_{ij} is the total number of utterances in cluster i, spoken by speaker j, n_i is the total number of utterances in cluster i, and N is the total number of utterances in the conversation.

The number of Gaussian components of the UBM and the GMM, were chosen as function of the conversation length. The longer the conversation, the higher the number of components. For typical conversation between 4-10 minutes we set 64 Gaussian components. For longer conversation (more then 10 minutes) we set 128 Gaussian components, while for very short conversations (less then 4 minutes) we set 32 Gaussian components.

3.4.2 Performance Evaluation

We evaluate the proposed speaker diarization system in comparison with four different methods: Baseline system, as described in detail in Section 2.3, PCA-based and LPP-based methods using GMM mean supervectors which are suggested by Hao et al. [13], and finally a different spectral clustering approach to speaker diarization [25]. Moreover, we investigated the graph embedding ability of spectral clustering in comparison with advanced algorithms such as NPE and LLE, in the perspective of speaker diarization. The same recordings tests were used in all implemented speaker diarization systems, while the VAD algorithm described in Chapter 4 is utilized only in the proposed speaker diarization system. Rest of the systems utilize a conventional VAD algorithm, which is proposed by Sohn et al [20].

This research addresses two primary issues: Short utterances diarization

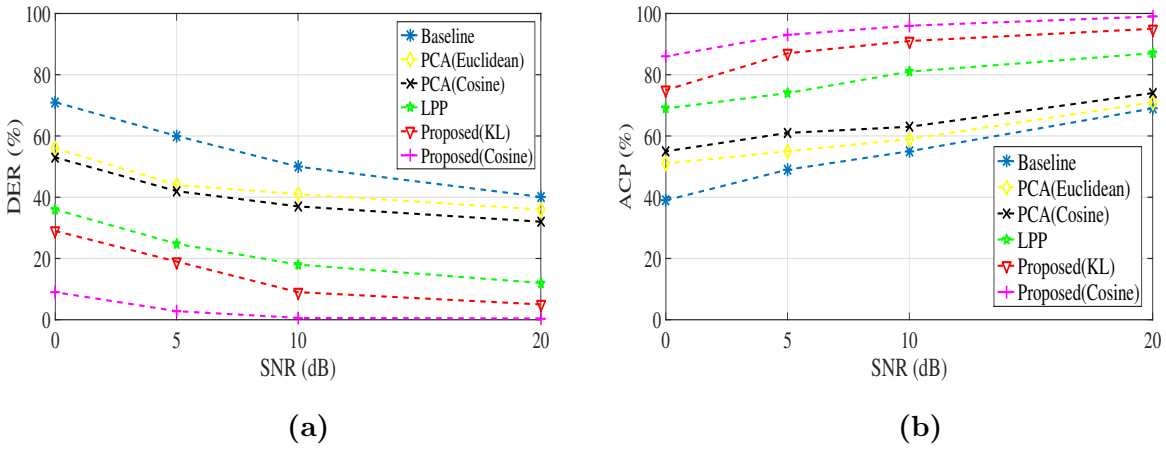
and noisy environments. First, we tested the proposed method on rapid speaker change point conversation corrupted by AWGN at SNR of 20dB. Table 3.1 summarizes the results.

Table 3.1: Performance Evaluation of Speaker Diarization Systems using Diarization Error Rate (DER) and Average Cluster Purity (ACP).

Method	Distance metric	DER (%)	ACP (%)
Baseline system	–	40.1%	68.2%
PCA (GMM mean supervector space)	Euclidean	36.1%	73.4%
	Cosine	25.6%	76.1%
LPP	Heat kernel	10.1%	89.1%
Proposed method	KL divergence	1.2%	97.5%
	Cosine	0.6%	99.5%

In the next stage, we added stationary noise at different levels of SNR as well as different transients. The results are summarized in Fig. 3.8 and Table 3.2.

Fig. 3.8 presents diarization performance in noisy environments include AWGN. Generally, the noise causes two main difficulties: First, extracting of relevant features as well as modeling speech segment are affected significantly by background noises. Second, pure segments of noise which are incorrectly considered as speech and take part in the modeling process, might also damage the representation reliability. Therefore, a significant performance degradation is observed.



(a) **(b)**

Figure 3.8: Diarization performance evaluation of two-speaker conversations corrupted by AWGN at various SNR levels. (a): DER. (b): ACP.

Table 3.2 demonstrates diarization performance in noisy conditions including babble noise at SNR level of 10dB and few types of transients. Because of the similarity to the desired target speech, speech babble is one of the most challenging noise interferences. It can be seen that the combination of babble noise with transients is equivalent, in term of performance (DER and ACP), to the case of using AWGN only at SNR level of 0-5dB. VAD algorithms suffer from a significant performance degradation in transient noise environments. The transients cause to high rates of false alarms and misses of speech frames. Due to the fact that those two errors are part of the DER calculation, VAD performance has significant effect on the total diarization system performance. When evaluating the diarization system performance in transient noise environments, we saw that false alarms and miss errors obtained by the proposed system are both about 15 percent. While using a conventional VAD algorithm, those errors increase to 35 percent and even more. In addition, the confusion errors is relatively low (about 4 percent). We also attribute the low confusion rate to our

approach for UBM training. We trained the UBM on the tested conversation, i.e., we made a kind of environment learning, which contributes to the MAP adaptation. When using traditional UBM training method, the DER increases in 3 percent at least.

Table 3.2: Performance Evaluation of Speaker Diarization Systems under Noisy Conditions Including Different Transients and Babble Noise at SNR Level of 10dB

Compared System	Door Knock	Keyboard Stroke	Metronome
Baseline	69.92%	71.23%	69.9%
PCA (Euclidean)	57.6%	56.2%	58.1%
PCA (Cosine)	56.8%	56.1%	57%
LPP	42.6%	43.9%	42.2%
Proposed (KL)	28.9%	27.2%	29.6%
Proposed (Cosine)	18.6%	19.3%	19.6%

As expected, the baseline system demonstrates the poorest performance. In case of short utterance, the number of extracted feature vectors is relatively low such that it is difficult to build a reliable and unique model for each segment. In addition, the noise also affects the models reliability, in particular transients, as described previously. Hence, during a speaker change point, the difference between two consecutive utterances is faded and the change point is missed. The false segmentation misleads the agglomerative clustering algorithm, which is also based only on models, and the whole system performance degrades. At low levels of SNR, the probability to misses and false alarms increases. The simulation demonstrates the essentiality of a novel VAD which is dedicated to handle with noisy conditions.

The PCA-based and LPP-based speaker diarization systems demonstrate better performance than the baseline system. One reason can be attributed to the VAD algorithm which enables fine segmentation and isolated one-speaker segments. However, the primary reason is the effective low-dimensional representation that captures the most important features of the data and discards noise and outliers. We found that LPP method is more suitable than PCA algorithm for speaker clustering task, and therefore demonstrates better results. In what follows, we discuss the LPP and PCA main characteristics and their effects on the speaker diarizarion performance.

LPP [31] is a linear dimensionality reduction algorithm which aims to preserve the local structure of the data in the original space. The LPP is obtained by finding the optimal linear approximations to the eigen-functions of the Laplacian-Beltrami operator on the manifold. Moreover, in comparison to PCA, the LPP algorithm is less sensitive to outliers because of its neighboring-preserving character [26]. PCA is an eigenvector method which is designed to model linear variations in a high-dimensional space. It performs dimensionality reduction by projecting the original data onto the reduced dimensional linear subspace, spanned by the leading eigenvectors of covariance matrix of the data. The objective of PCA is to find a set of orthogonal basis functions that hold the directions of maximum variance. It effectively identifies only the Euclidean structure, i.e., it aims to preserve the global structure of the data and fails to discover the underlying structure, the local structure. In addition, the PCA is restricted only to convex and linear manifolds. The fact that the distribution of natural

data, like speech, is non-uniform and is not necessarily convex or linear, as well as can be represented in low-dimensional structures, motivates us to exploit the shape geometry of the distribution and encourages the use of embedding algorithms such as LPP and spectral clustering approaches.

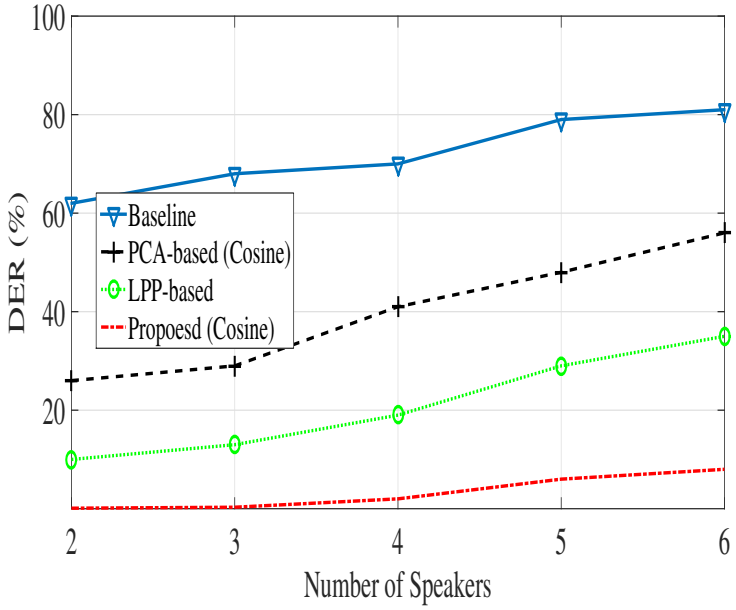


Figure 3.9: Performance evaluation of proposed method in comparison with compared method under different number of speakers involved in the conversation. As the number of speakers increases, the performance are degraded.

When increasing the number of speakers, the discrimination between different speakers becomes harder and the diarization performance is degraded. It is clearly demonstrated in Fig.3.9 that the spectral clustering approach is less sensitive to the number of speakers parameter. After the number of speakers (K) is estimated, the first K eigenvectors are chosen, i.e., the most informative data is captured without dependency at number of speakers.

We also evaluated the performance of the proposed method in comparison with Ning et al. [25] diarization system, which is also based on spectral

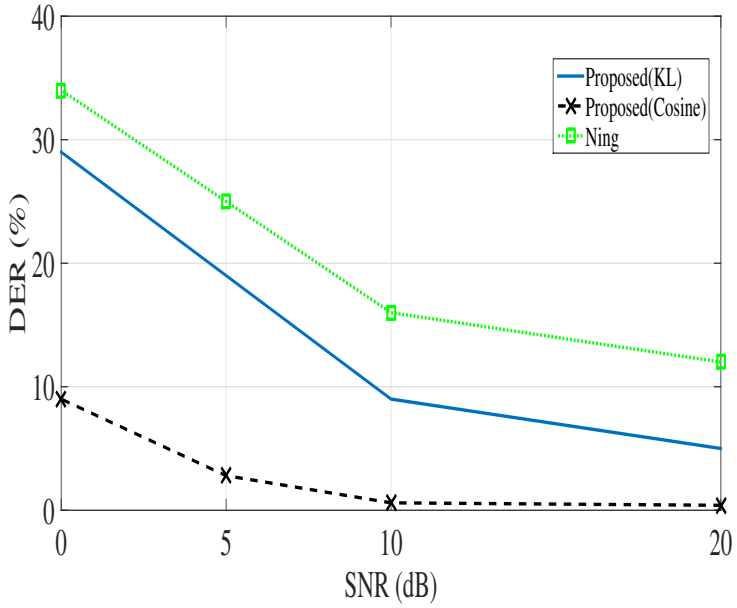


Figure 3.10: Diarization Error Rate of the proposed method in comparison with Ning et al. diarization system which is also based on spectral clustering.

clustering. Fig. 3.10 demonstrates that the proposed diarization system outperforms the compared method. The following milestone stages of the proposed algorithm significantly contribute to the diarization performance. First, using a unique VAD for fine segmentation instead of common BIC-based speaker change point detector provides better segmentation which suits to short utterances diarization in noisy environments. Second, exploiting the GMM supervector and its first and second derivatives. Using the first and second derivatives reduced the total DER by 6 percent, averagely. This phenomenon can be explained by the informative data which hidden on GMM mean supervectors trajectories over time. It emphasizes the dissimilarity between consecutive utterances of different speakers. Third, the choice of cosine metric which improves the similarity matrix by enhancing the connection between relevant utterances.

Another interesting view for evaluating dimensionality reduction algo-

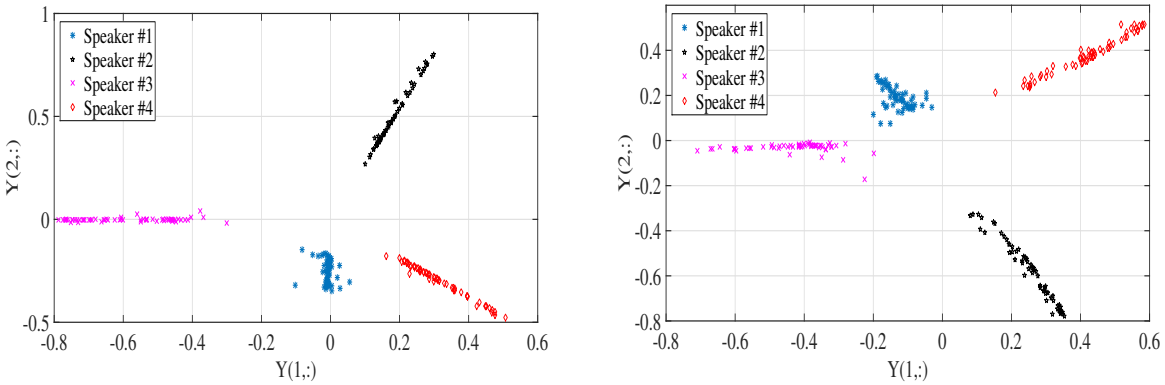


Figure 3.11: Scatter plots of the new representation. The tested conversation includes 240 utterances (data points) corrupted by AWGN at different SNR levels and keyboard stroke. Each colour represents a specific speaker. (a): SNR of 10dB. (b): SNR of 5dB.

rithms is scatter plot which demonstrates the low-dimensional representation. Usually, where the data points are concentrated around clear centers such as each group is isolated from the others, a conventional clustering algorithms like k-means can provide a perfect clustering. For example, in Fig.3.11 the data is scattered clearly to four groups i.e., four speakers. Each data point represents an utterance and each color represents different speaker. The supervectors are transformed to a reduced subspace which is comprised of the most informative eigenvectors of the Laplacian matrix and enables perfect clustering. It is worth to mention that for convenient visualization, we projected the data on 2D sub-space rather on 4D sub-space (i.e., the first two eigenvectors instead of the first four).

Focusing on LPP and PCA graph embedding performances, it can be clearly seen from Fig. 3.12 that the LPP introduces better results than the PCA. However, it is also outstanding that the LPP has performance degradation in case of low SNR levels; the scatter plot which corresponds

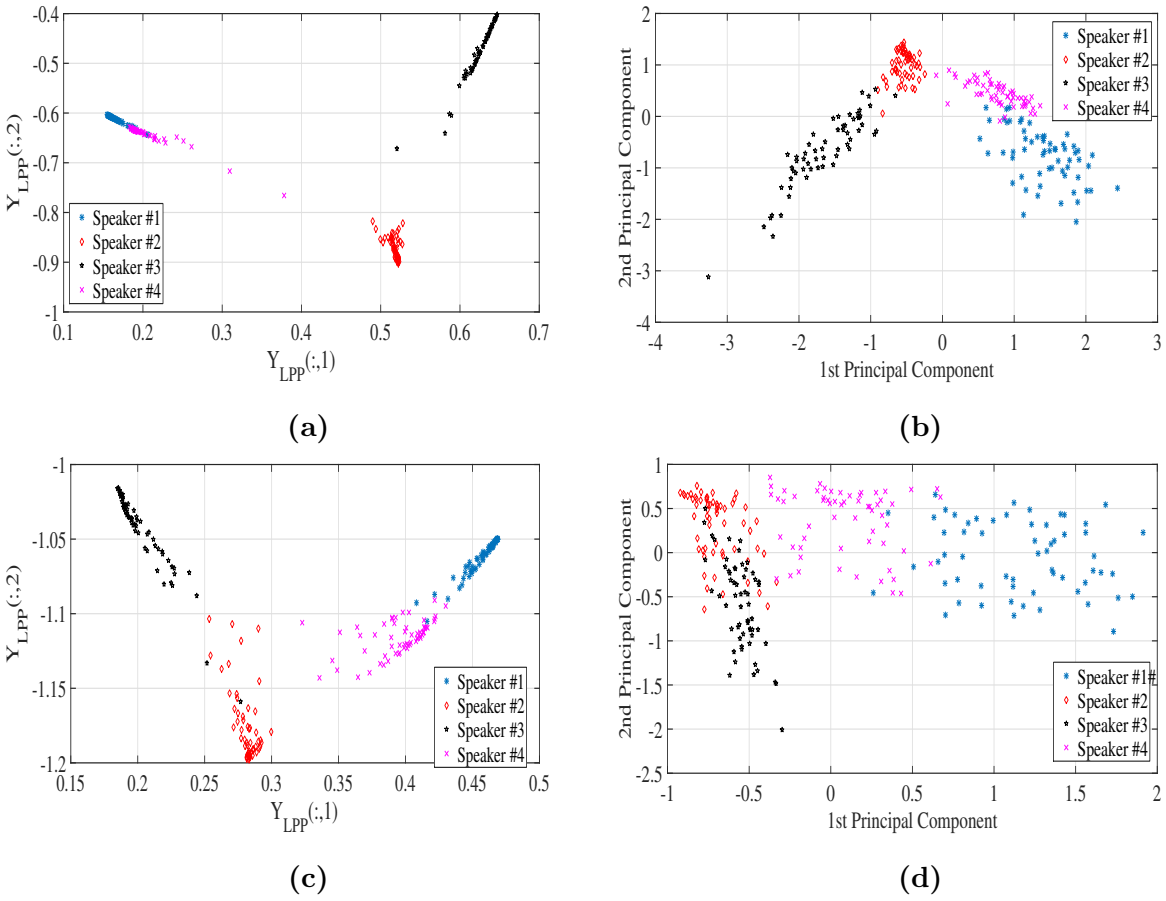
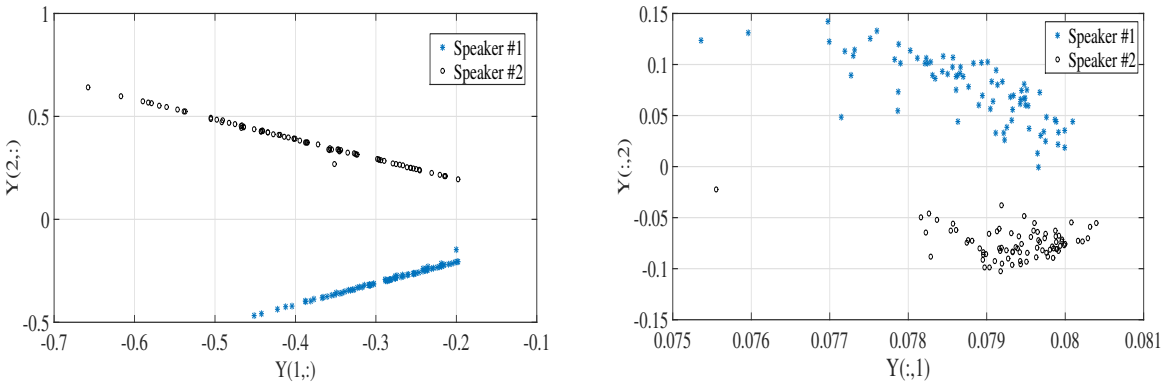


Figure 3.12: Scatter plots of new representation using LPP and PCA methods. The conversation composed of 240 utterances (data points) corrupted by AWGN at different levels of SNR. $\boldsymbol{Y}_{LPP}(:,1)$ and $\boldsymbol{Y}_{LPP}(:,2)$ are a notation for the first and second axes of the reduced subspace obtained by LPP. (a): LPP at SNR of 10dB. (b): PCA at SNR of 10dB. (c): LPP at SNR of 5dB. (d): PCA at SNR of 5dB.

with SNR of 10dB demonstrates better separation than the one which corresponds with to SNR of 5dB.

When comparing the distance metrics, it can be deduced that the cosine metric shows better results than KL divergence. The KL divergence measures distance between probability distributions of data points. Hence, we used the GMM mean representation without the first and second derivatives as proposed. Moreover, Fig. 3.13 demonstrates the scattering pattern



(a) **(b)**

Figure 3.13: Scatter plots of the new representation created by spectral clustering method. The example includes two-speaker conversation corrupted by AWGN at SNR of 10dB. (a): Using cosine metric. (b): Using KL divergence.

of the data in the resulted space using spectral clustering. It is clearly seen that when using the cosine metric, the data is more concentrated and the clusters boundaries are well defined.

The proposed speaker diarization system shows significant improvement in ACP and DER measures of performances in comparison with competing speaker diarization systems. The excellent results of the proposed system are derived from the combination of three primary factors. The first one is the fine segmentation. Using a VAD algorithm instead of BIC-based speaker change point detector contributes to divide the speech sequence into isolated segments which consist of one speaker only. Second, the supervector representation enables us to treat each utterance as a specific data point and supply conventional scattering pattern. Furthermore, this method enables a successful separation between different speakers even in cases of limited extracted data i.e., short utterances. The third factor is the spectral clustering that has two specific advantages which tip the scales

in its favour: it is not limited to linear constraints and it preserves the local structure of the data which is necessary, as mentioned before. Besides, in the original form, the data points do not lie on linear manifolds, and using a non-linear algorithm is more suitable. Therefore, using spectral clustering demonstrates better results when comparing with LPP or PCA approaches.

In order to challenge the proposed algorithm, we also investigated its performance in comparison with neighborhood preserving embedding (NPE) and locality linear embedding (LLE) algorithms [27, 31], which are advanced dimensionality reduction methods. NPE and LLE are based on minimizing the reconstruction errors, i.e., it assumes that each data point can be reconstructed from its neighbors (see Chapter 2, Section 2.4.5 and 2.4.2). Table 3.3 presents the performance evaluation of LLE-based and NPE-based diarization systems. The results are averaged on all recording files which include 2-6 speakers. Those systems hold the same stages as the proposed diarization system except of the utilized dimensionality reduction method. Instead of using spectral clustering, LLE or NPE are used and followed by k-means clustering. While investigating the LLE algorithm,

Table 3.3: Diarization Performance Evaluation of The Proposed System in Comparison with LLE-based and NPE-based Diarization Systems under Stationary Noise at SNR Level of 0dB.

Compared System	ACP
LPP	72%
NPE	73.3%
LLE	79.2%
Proposed (Cosine)	86%

we recognized that the performance highly depends on the parameter D, which is the reduced dimension. Choosing a value which is higher than the number of speakers causes to significant performance degradation. In the following example 3.14, we applied LLE algorithm for different values of D. The figure shows that the clusters are less defined when D is greater than the known number of clusters. Regard the diarization performance, the ACP decreases in more then 25 percent when choosing non-suitable value for D. In case of short conversations, the LLE and NPE performances are

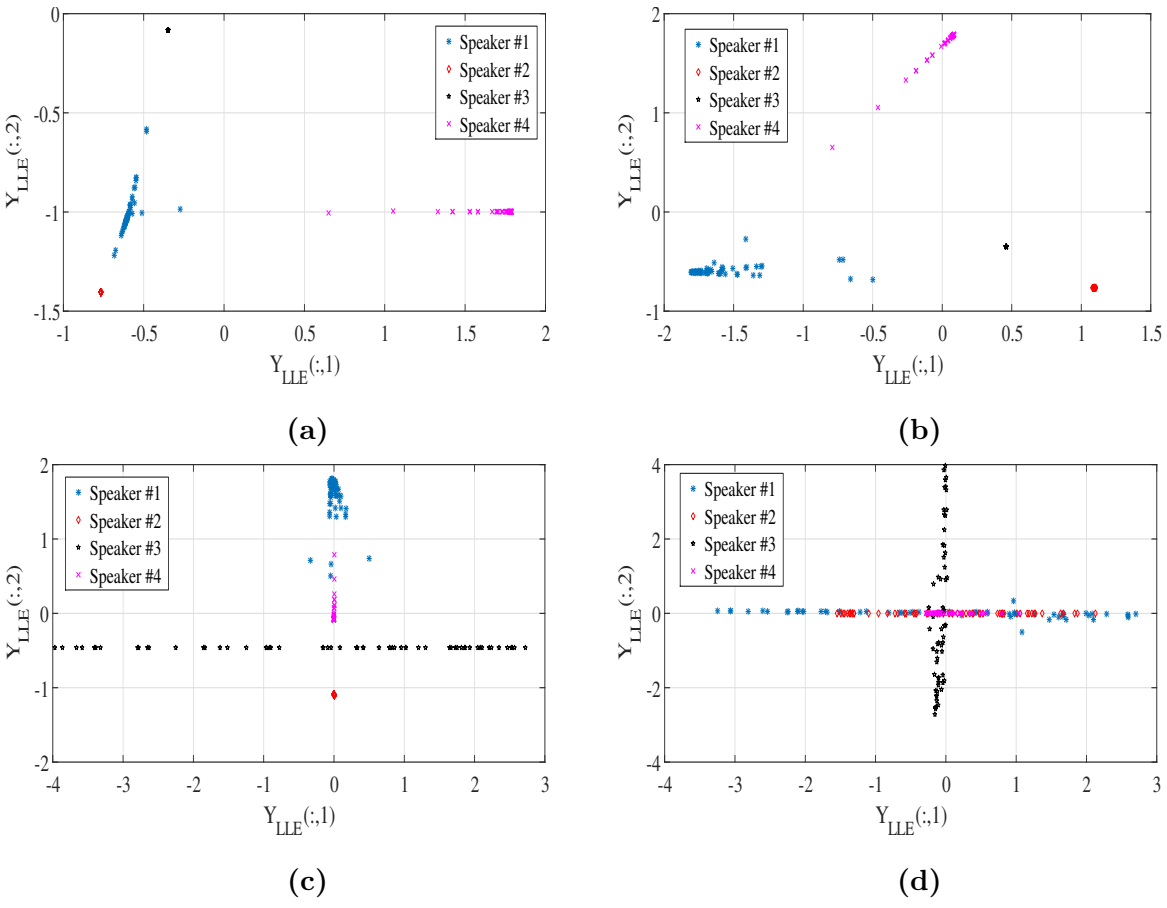


Figure 3.14: Scatter plots of the new representation which is obtained by LLE algorithm. Each plot demonstrates the new representation for different value of D. $\boldsymbol{Y}_{LLE}(:,1)$ and $\boldsymbol{Y}_{LLE}(:,2)$ are a notation for the first and second axes of the reduced subspace obtained by LLE. (a): D = 3. (b): D = 4. (c): D = 5. (d): D = 6.

degraded. The scatter plots presented in Fig.3.15c and Fig.3.15d demonstrate very distributed scattering patterns. However, it can be also seen that using the spectral clustering provides isolated groups which lead to perfect clustering using k-means algorithm. The NPE and LLE both aim

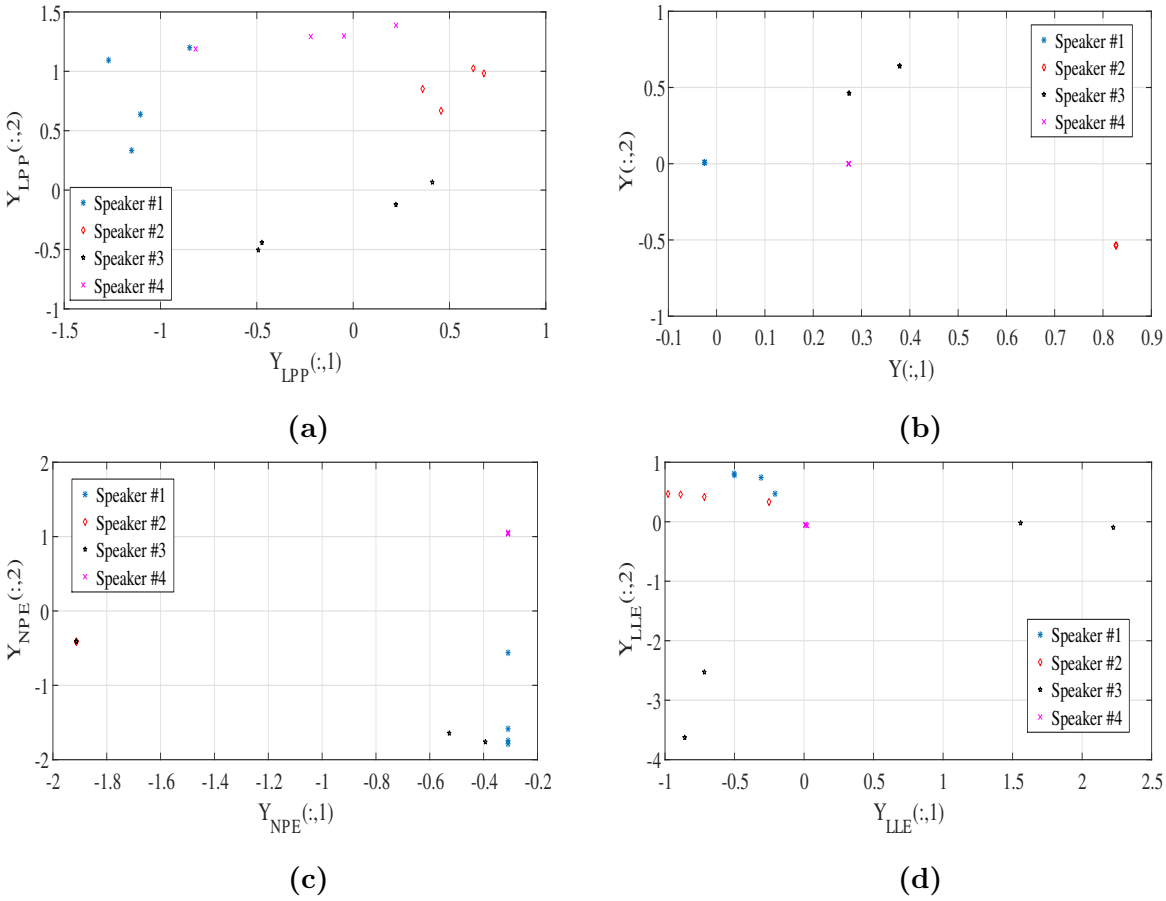


Figure 3.15: Scatter plots of the new representations obtained by LPP, spectral clustering, NPE and LLE, respectively. The conversation composed of 16 utterances (data points) under SNR level of 0dB. $\boldsymbol{Y}_{NPE}(:,1)$ and $\boldsymbol{Y}_{NPE}(:,2)$ are a notation for the first and second axes of the reduced subspace obtained by NPE. (a): LPP. (b): Spectral clustering. (c): NPE. (d): LLE.

to minimize the same cost function which is based on locally linear reconstruction errors. They holds few assumptions: First, the manifold is comprised of linear patches, and second it is well-sampled such that each data point can be reconstructed from its neighbors. Therefore, applying these

algorithms on non-uniform sample manifold or on well-distributed densities leads to significant performance degradation. However, when enough utterances are produced from each speaker, this criterion is suitable. It preserve also local global connections which can also be helpful in handling with noisy environments.

It should be emphasized that the last stage before final diarizarion is applying k-means on the reduced-dimension vectors. As long as the data points will be concentrated around a specific point, the k-means algorithm will produce better solution. Our research shows that spectral clustering approach for speaker diarization provides impressive performances. It introduces the idea that preserving the local structure of the data is required. In addition, it shows that linearity is not always a desirable property, and there are some cases in which the restriction to linearity degrades the performance.

3.5 Conclusions

We have proposed a speaker diarization algorithm which is based on GMM mean supervectors and spectral clustering. Our main concern was dealing with the challenging cases of noisy environments and short utterances i.e., rapidly speaker change points. Our speaker diarization algorithm is an off-line unsupervised algorithm. It requires the tested conversation in advanced in order to calculate the general speaker model, the UBM. We trained the UBM on the noisy tested conversation. We gained an improvement of performances in noisy environments, less Gaussian components and better discrimination between different speakers. Each utterance was represented by GMM supervectors. This representation enables to map a segment of speech to a high dimensional vector which is a great platform to any graph embedding algorithm, and in particular spectral clustering. Furthermore, the MAP adaptation causes shifts in local modes and regional centers of mass. Hence, the adapted GMM mean supervector actually represents local first-order differences between the UBM and the adapted GMM, i.e., the discrimination between difference speakers is emphasized. Regard the spectral clustering method, we have examined two types of metric distances: KL divergence and cosine metric, as probability-based and vector-based distances, respectively. These metrics distances are required for constructing the similarity matrix which is part of the spectral clustering. We noticed that the cosine metric provides better results then the KL divergence thanks to two main reasons: First, the derivatives of supervectors which were only used in the cosine metric and second, the neglected

magnitude which are part of the cosine metric properties. We compared our method to advanced dimensionality reduction methods like LLE, LPP, and NPE. The proposed algorithm demonstrates the best results. We also demonstrated that the linear approximation of non-linear techniques like LLE and spectral clustering, degrades the clustering performance. Noisy environments including highly non-stationary and transient noises are a great challenges which degrade the performances of each speaker diarization. In particular, when evaluating the algorithms by DER measure. As can be seen in (3.23), the VAD performance affects on two terms, the missing time T_M and the false alarm time T_{FA}. In order to handle with this challenge, we implemented a unique VAD algorithm which focuses on detecting speech frames in presence of very noisy environments. In case of relatively clean environment, conventional VADs may be satisfactory.

Chapter 4

Voice Activity Detection in Presence of Transient Noise

4.1 Introduction

VAD is required in many speech communication applications, such as speech recognition, speech coding, hands-free telephony, speech enhancement and echo cancellation. As discussed in Section 2.2, when segments of noise are incorrectly detected as speech segments, the feature extraction and modeling performance are degraded and so is the speaker representation.

Therefore, combining improved VAD which can detect speech in noisy environments is a real necessity. Elementary VAD algorithms rely on averaged parameters over frames such as zero crossing rate, pitch period, autocorrelation coefficients, energy levels, etc. These methods have applicable results for clean speech signals, but in noisy environments their performance severely degrades. To overcome this shortcoming, several statistical model based VAD algorithms have been proposed in the last two decades. Sohn et al. [20] assumed that the spectral coefficients of

the noise and speech signal can be modeled as complex Gaussian random variables, and developed a VAD algorithm based on the LRT. Following their work, researchers tried to improve the performance of model-based VAD algorithms by assuming different statistical models for speech signals, see [16,17,20,50–52]. While these methods perform well in stationary noisy environments, as long as the SNR is not too low, their performance degrades significantly in presence of transient noise. Transient noise are poorly modeled by statistical models and does not slowly varying with respect to speech. Therefore, state-of-the-art model-based VAD are less suitable in this case.

In this section, we present a new supervised learning VAD algorithm which is based on the Laplacian pyramid algorithm. Training data is used for estimating the likelihood ratio function and calculating required parameters for the Laplcaian pyramid algorithm. The training data is also used in finding two Gaussian mixture models for modeling the first two eigenvectors of the Laplacian of the similarity matrix corresponding to the first two leading eigenvalues of the normalized Laplacian matrix. Upon receiving new unlabeled data, the Laplacian pyramid algorithm is used for evaluating the likelihood ratio. The final VAD is obtained by comparing that likelihood ratio to a threshold. More specifically, let $\mathcal{H}_1$ be the hypothesis of speech presence and $\mathcal{H}_0$ be the hypothesis of speech absence. The test is defined as:

$$LR = \frac{Pr(\boldsymbol{x}_t; \mathcal{H}_1)}{Pr(\boldsymbol{x}_t; \mathcal{H}_0)} \lessgtr Th. \tag{4.1}$$

If the LR (likelihood-ratio) is greater then a threshold, the frame contains speech and vice versa.

The rest of this section is organized as follows. In Section 4.2, we formulate problem of voice activity detection in noisy environments. In Section 4.3, we briefly discuss the Laplacian pyramid algorithm which is used here as function extension tool, and in Section 4.4, we introduce our proposed VAD algorithm. Simulation results and performance evaluation are presented in Section 4.5. Finally, we conclude this chapter in Section 4.6.

4.2 Problem Formulation

Let $x_{\text{sp}}(n)$ denote a speech signal and let $x_{\text{tr}}(n)$ and $x_{\text{st}}(n)$ denote the additive contaminating transient and stationary noise signals, respectively. The signal measured by a microphone is given by:

$$y(n) = x_{\text{sp}}(n) + x_{\text{tr}}(n) + x_{\text{st}}(n). \tag{4.2}$$

The noisy signal $y(n)$ is divided into time frames of 32msec each while the goal is to determine whether a given frame contains speech or not. The algorithm output is a vector of indices which is defined as follows:

$$\mathbb{1}_t = \begin{cases} 1 & \text{if frame } t \text{ contains speech signal;} \\ 0 & \text{if frame } t \text{ contains non-speech signal.} \end{cases}$$

4.3 Laplacian Pyramid Algorithm

The Laplacian pyramid is a multi-scale algorithm for extending any function f, which is defined on data set S, to new data set of points. The function f is approximated in different resolutions, using mutual distances between the data points in S. The Laplacian pyramid algorithm has been

used in some applications. Burt and Adelson [55] introduced the Laplacian pyramid for image coding. Rabin and Coifman [56] proposed to represent and learn heterogeneous data sets using diffusion maps for embedding and Laplacian pyramid representation, which extend the embedding to new data set. The authors demonstrate the benefit of Laplacian pyramid as function extension algorithm, in comparison with geometric harmonics. In the following section we briefly introduce the Laplacian pyramid algorithm. Let S be a set of data points in $\mathbb{R}^m$, the similarity matrix between the data points in the dataset is determined as:

$$\boldsymbol{W} = w(\boldsymbol{x}_i, \boldsymbol{x}_j) = exp(-||\boldsymbol{x}_i - \boldsymbol{x}_j||^2/\sigma). \tag{4.3}$$

In order to simplify the description of the algorithm, we use the Gaussian kernel but any other kernel can be taken into consideration. Yet, we define this matrix for different resolutions. The first resolution is corresponding to the regular kernel as is defined in (4.3):

$$\boldsymbol{W}_0 = w_0(\boldsymbol{x}_i, \boldsymbol{x}_j) = exp(-||\boldsymbol{x}_i - \boldsymbol{x}_j||^2/\sigma). \tag{4.4}$$

Normalizing by $\sum_j w_0(\boldsymbol{x}_i, \boldsymbol{x}_j)$ yields a smoothing operator $\boldsymbol{K}_0$ which is defined as:

$$\boldsymbol{K}_0 = k_0(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{w_0(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\sum_j w_0(\boldsymbol{x}_i, \boldsymbol{x}_j)} \tag{4.5}$$

At a finer scale, l, the kernel is defined as:

$$\boldsymbol{W}_l = w_l(\boldsymbol{x}_i, \boldsymbol{x}_j) = exp(-||\boldsymbol{x}_i - \boldsymbol{x}_j||^2/(\frac{\sigma}{2^l})) \tag{4.6}$$

and the smoothing operator is defined accordingly as:

$$\boldsymbol{K}_l = k_l(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{w_l(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\sum_j w_l(\boldsymbol{x}_i, \boldsymbol{x}_j)} \tag{4.7}$$

Then, the Laplacian pyramid representation of function f is given as follows. The first level ($l = 0$) is defined as:

$$\boldsymbol{s}_0(\boldsymbol{x}_k) = \sum_{i=1}^{n} K_0(\boldsymbol{x}_i, \boldsymbol{x}_k) f(\boldsymbol{x}_i) \tag{4.8}$$

and for any other level ($l = 1, 2, ..., v$) is defined as:

$$\boldsymbol{s}_l(\boldsymbol{x}_k) = \sum_{i=1}^{n} K_l(\boldsymbol{x}_i, \boldsymbol{x}_k) d_l(\boldsymbol{x}_i). \tag{4.9}$$

The differences d_l, are calculated as:

$$d_{\mathrm{l}} = f - \sum_{i=0}^{l-1} \boldsymbol{s}_{\mathrm{i}}. \tag{4.10}$$

The iterations in (4.8)-(4.9) stop when $\|f - \sum_k \boldsymbol{s}_{\mathrm{k}}\| < Error$, which is *a-priori* defined. The extension of function f to an unlabeled data point z is calculated by:

$$\boldsymbol{s}_0(\boldsymbol{z}) = \sum_{i=1}^{n} K_0(\boldsymbol{x}_i, \boldsymbol{z}) f(\boldsymbol{x}_i) \tag{4.11}$$

$$\boldsymbol{s}_l(\boldsymbol{z}) = \sum_{i=1}^{n} K_l(\boldsymbol{x}_i, \boldsymbol{z}) d_l(\boldsymbol{x}_i). \tag{4.12}$$

Finally, the value of function f for an unlabeled data point z is calculated as:

$$f(\boldsymbol{z}) = \sum_{k=0}^{l-1} \boldsymbol{s}_{\mathrm{k}}(\boldsymbol{z}). \tag{4.13}$$

4.4 Proposed Algorithm

Our proposed algorithm is a supervised learning algorithm which means that someone has to train it before using. The final decision regard speech presence or absence is based on likelihood ratio test. The likelihood ratio

are calculated by extension the likelihood ratio function of training data using the Laplacian pyramid algorithm. More specifically, let the function Γ^{train} be the likelihood ratio function on the training database. Our goal is to extract this function to an unlabeled frame time z, i.e., $\Gamma_{\text{z}}^{\text{test}}$, and compare it to a given threshold for deciding on speech presence or absence.

4.4.1 Feature Extraction

In the proposed VAD algorithm, we choose two types of features. The first feature is the absolute value of MFCCs, as is used in the utterance representation stage, see Section 3.3.2. Speech activity detection is an easier problem than discrimination between speakers and therefore we can satisfy on MFCC only, without its derivatives. The second chosen feature is the arithmetic mean of the log-likelihood ratio for the individual frequency bins.

More specifically, let $\boldsymbol{Y}_{\text{m}}(k,t); (t = 1, ..., N; k = 1, 2, ..., K_s)$ and $\boldsymbol{Y}_{\text{s}}(k,t); (t = 1, ..., N; k = 1, 2, ..., K_m)$ be the absolute value of the MFCCs and short time Fourier transform (STFT) coefficients, respectively. K_m and K_s are respectively the frequency bins which the MFCC and STFT coefficients are computed in. The feature vector of frame t is defined as the column concatenation of those two components as follows:

$$\boldsymbol{Y}(:,t) = \begin{bmatrix} \boldsymbol{Y}_{\text{m}}(:,t) \\ \Lambda_{\text{t}} \end{bmatrix} \tag{4.14}$$

where $\boldsymbol{Y}_{\text{m}}(:,t)$ is the absolute value of MFCCs in frame t and Λ_{t} is the arithmetic mean of the log-likelihood ratio for the individual frequency bands in frame t. Assuming that the signal and the noise both have uncorrelated

Gaussian distribution in the STFT domain, the arithmetic mean of the log-likelihood ratio for the individual frequency bands in frame t can be formulated as follows:

$$\Lambda_t = \frac{1}{K_s} \sum_{k=1}^{K_s} \left(\frac{\gamma_k(t)\xi_k(t)}{1+\xi_k(t)} - \log\left(1+\xi_k(t)\right) \right) \tag{4.15}$$

The variables $\xi_k(t)$ and $\gamma_k(t)$ are called *a-priori* SNR and *a posteriori* SNR, respectively and are defined as follows:

$$\xi_k(t) = \lambda_s(t,k)/\lambda_N(t,k) \tag{4.16}$$

$$\gamma_k(t) = |\boldsymbol{Y}_s(t,k)|^2/\lambda_N(t,k) \tag{4.17}$$

where $\lambda_s(t,k)$ is the variance of speech signal in the k-th frequency bin of the t-th frame and $\lambda_N(t,k)$ is the variance of stationary noise in the k-th frequency bin of the t-th frame. The *a-priori* SNR can be estimated using decision-directed method [53] and $\lambda_N(t,k)$ can be estimated from training data, assuming regions of only stationary noise in the sequence or by improved minima controlled recursive averaging (IMCRA) [54].

The likelihood ratio has been long exploited as a feature for voice activity detection in presence of stationary noise [16, 17, 20, 50, 51]. In addition, MFCCs are commonly used as features in speech and speaker recognition systems. Therefore, combining these two features appropriately, would be a suitable feature space for voice activity detection in noisy environments [19, 21].

4.4.2 Training Stage

The training stage has two main goals. First, calculating the likelihood ratio function of the training data and second, extracting essential param-

eters which are required for the Laplacian pyramid representation.

Suppose that we have a database of clean speech signal, a database of transient noise and a database of stationary noise. We choose L different signals from each database and combine them. Let $x_{\rm sp}^{\ell}(n), x_{\rm tr}^{\ell}(n), x_{\rm st}^{\ell}(n)$ be the ℓ-th speech signal, transient noise, and stationary noise, respectively. Without loss of generality, we assume that all of these signals are of the same length (i.e. N_ℓ). Yet, we build the ℓ-th training sequence, $\boldsymbol{Y}^{\ell}$, as follows:

$$x_1^{\ell}(n) = x_{\rm sp}^{\ell}(n) + x_{\rm st}^{\ell}(n), \tag{4.18}$$

$$x_2^{\ell}(n) = x_{\rm tr}^{\ell}(n) + x_{\rm st}^{\ell}(n), \tag{4.19}$$

$$x_3^{\ell}(n) = x_{\rm sp}^{\ell}(n) + x_{\rm tr}^{\ell}(n) + x_{\rm st}^{\ell}(n), \tag{4.20}$$

Let $\boldsymbol{Y}_1^{\ell}$, $\boldsymbol{Y}_2^{\ell}$, $\boldsymbol{Y}_3^{\ell}$ be the feature matrix extracted using (4.14) from $x_1^{\ell}(n)$, $x_2^{\ell}(n)$, $x_3^{\ell}(n)$, respectively, and $\boldsymbol{Y}^{\ell}$ be the row concatenation of these matrices as follows:

$$\boldsymbol{Y}^{\ell} = \left[\boldsymbol{Y}_1^{\ell}{:}\boldsymbol{Y}_2^{\ell}{:}\boldsymbol{Y}_3^{\ell}\right]. \tag{4.21}$$

The fact that distribution of speech signal and transient noise can concentrate around low dimensional structures [6], motivates us to underly the structure of the data in lower dimensional space. In the proposed method we choose spectral clustering for this purpose. Spectral clustering is widely discussed in 2.4.3. In general, the main objective of spectral clustering is to exploit the geometry of the distribution by using the leading eigenvectors of the Laplacian of the similarity matrix as the new low dimension representation of the data. The most important issue in every kernel based method like spectral clustering, is defining an appropriate kernel which

preserves the similarity between points. Here, we define the parametric kernel as follows:

$$\boldsymbol{W}_{\theta}^{\ell}(i,j) = \exp\left(\sum_{p=-P}^{P} -\alpha_{\mathrm{p}}\boldsymbol{Q}(i+p,j+p)\right) \tag{4.22}$$

$$\begin{aligned}\boldsymbol{Q}(i,j) = \| &\boldsymbol{Y}_{\mathrm{m}}^{\ell}(:,i)\left(1-\exp(-\Lambda_{\mathrm{i}}^{\ell}/\epsilon)\right) - \\ &\boldsymbol{Y}_{\mathrm{m}}^{\ell}(:,j)\left(1-\exp(-\Lambda_{\mathrm{j}}^{\ell}/\epsilon)\right)\|_2^2\end{aligned} \tag{4.23}$$

where $\boldsymbol{\theta} = [\epsilon, \alpha_{-P}, \alpha_{-P+1}, \cdots, \alpha_{P-1}, \alpha_P] \in \mathbb{R}^{2P+2}$ is a vector of parameters. $\boldsymbol{Y}_{\mathrm{m}}^{\ell}(:,i)$ is the absolute value of the MFCCs, and $\Lambda_{\mathrm{i}}^{\ell}$ is the arithmetic mean of log likelihood ratios for the individual frequency bands of the ℓ-th training sequence, in frame i. Two main considerations are taken into account in choosing the described weight matrix: The similarity between two individual frames and the effect of neighboring frames, which is characterized by the usage of P consecutive frames from each side. When the current frame contains speech or transient, the likelihood feature is relatively high. Therefore, it can be deduced from (4.22-4.23) that the value of the exponential term in (4.23) is around zero and the term $\left(1-\exp(-\Lambda_{\mathrm{j}}^{\ell}/\epsilon)\right)\|_2^2$ tends to 1. In this case, the similarity matrix depends only on the MFCCs which are indeed relevant for speech and transient noise only. Therefore, combining these two features of MFCCs and likelihood ratios in the proposed kernel which is described in (4.22)-(4.23), leads to a suitable metric that can be utilized as a similarity notion between two frames for VAD in noisy environments [19]. The kernel parameters are obtained by solving

the following optimization problem [24]:

$$\boldsymbol{\theta}^{opt} = \arg\min_{\theta} \frac{1}{L} \sum_{\ell=1}^{L} F(\boldsymbol{W}_{\boldsymbol{\theta}}^{\ell}, \boldsymbol{C}^{\ell}) \tag{4.24}$$

$$F(\boldsymbol{W}, \boldsymbol{C}) = \frac{1}{2} \left\| \boldsymbol{\Upsilon}\boldsymbol{\Upsilon}^T - \boldsymbol{D}^{1/2}\boldsymbol{C}(\boldsymbol{C}^T\boldsymbol{D}\boldsymbol{C})^{-1}\boldsymbol{C}^T\boldsymbol{D}^{1/2} \right\|_F^2$$

where L is the number of training sequences, $(\cdot)^T$ denotes transpose of a vector or a matrix, $\boldsymbol{\Upsilon}$ is an approximate orthonormal basis of the projections on the second principal subspace of $\boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$ obtained by classical *orthogonal iteration* [57]. In practice we use the gradient method (e.g. *fminunc* or *fmincon* functions in Matlab if there exists any constraint on the parameters) to solve this minimization problem.

Let $\boldsymbol{W}^{\ell}$ be the similarity matrix of the ℓ-th training sequence, $\boldsymbol{D}$ is the degree matrix, and $\boldsymbol{U}_{\ell} \in \mathbb{R}^{3N_{\ell} \times 2}$ be a matrix consisting of the two eigenvectors of $\boldsymbol{D}^{\ell^{-1/2}}\boldsymbol{W}^{\boldsymbol{\ell}}\boldsymbol{D}^{\ell^{-1/2}}$ corresponding to the first two largest eigenvalues. In fact, the rows of matrix $\boldsymbol{U}_{\ell}$ are the new representation of the feature vectors, when each row represents a frame of $y(n)$. We chose the first two eigenvectors because of two reasons. Empirically, the third eigenvector has less contribution to the manifold representation and second, there is a clear connection between the number of clusters and the spectrum of eigenvalues (see 3.3.7). Suppose we have L training sequences, let the column concatenation of $\boldsymbol{U}_1$ through $\boldsymbol{U}_L$ be $\boldsymbol{U}$. In fact, the matrix $\boldsymbol{U}$ is a new representation of the training data such that each row of $\boldsymbol{U}$ corresponds to a specific training frame. Yet, after obtaining the new representation of the training sequences, we use Gaussian mixture modeling to model each cluster (i.e. speech presence or absence) with a different GMM. For each cluster we find the rows of the matrix $\boldsymbol{U}$ corresponding to that cluster using

an indicator vector. Then, by exploiting the EM algorithm and BIC, we fit GMMs to those clusters. This means that we model the low dimensional representation of the original data using two different GMMs, one for each cluster. The likelihood ratio for each labeled frame t is obtained by:

$$\Gamma_{\mathrm{t}}^{\mathrm{train}} = \frac{f(\boldsymbol{U}(t,:);\mathcal{H}_1)}{f(\boldsymbol{U}(t,:);\mathcal{H}_0)} \tag{4.25}$$

where $\boldsymbol{U}(t,:)$ is the t-th row of the matrix $\boldsymbol{U}$, and $\mathcal{H}_1$ and $\mathcal{H}_0$ are the speech presence and absence hypotheses, respectively.

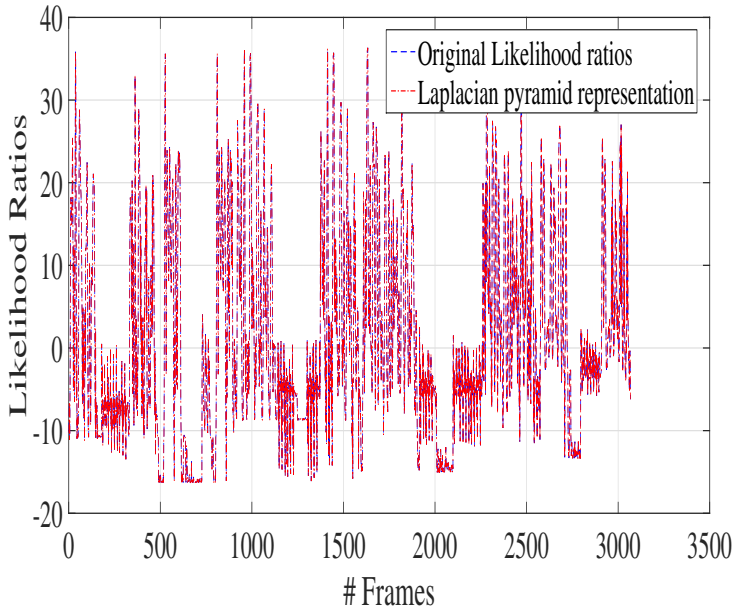


Figure 4.1: Likelihood ratio functions: Estimated by the Laplacian pyramid algorithm (in red) and the calculated function (in blue).

Finally, we would like to represent the likelihood ratio function (LRF) by the Laplacian pyramid algorithm. As it has already mentioned, the Laplacian pyramid is a multi-scale algorithm for extending an empirical function f, which is defined on a dataset S, to new data points. In our case, this function is the LRF of the training frames ($\boldsymbol{\Gamma}^{\mathrm{train}}$), and the dataset S is the collection of the labeled feature vectors. A general description of

the Laplacian pyramid algorithm is provided in 4.3. In what follows, we describe the use of the Laplacian pyramid algorithm within the proposed algorithm. First, we represent the training LRF in different resolutions which are approximated by mutual distances between the data points. In order to emphasize the mutual distances between the feature vectors we use the same kernel as formulated in (4.22)-(4.23). The similarity matrix for each resolution level $v = 0, 1, ..$ is given by:

$$\begin{aligned} \boldsymbol{W}_{\theta}^{\mathrm{v}}(i,j) &= \exp\left(\sum_{p=-P}^{P} -\alpha_{\mathrm{p}}\boldsymbol{Q}^{\mathrm{v}}(i+p,j+p)\right) \qquad (4.26) \\ \boldsymbol{Q}^{\mathrm{v}}(i,j) &= 2^{v}\|\boldsymbol{Y}_{\mathrm{m}}(:,i)\,(1-\exp(-\Lambda_{\mathrm{i}}/\epsilon)) - \\ &\qquad \boldsymbol{Y}_{\mathrm{m}}(:,j)\,(1-\exp(-\Lambda_{\mathrm{j}}/\epsilon))\,\|_2^2 \qquad (4.27) \end{aligned}$$

and the smoothing operator $\boldsymbol{K}_{\mathbf{v}}(i,j)$ is defined by:

$$\boldsymbol{K}_{\mathrm{v}}(i,j) = k_{\mathrm{v}}(i,j) = \frac{w_{\theta}^{\mathrm{v}}(i,j)}{\sum_{j=1}^{n} w_{\theta}^{\mathrm{v}}(i,j)} \qquad (4.28)$$

where n is the number of training frames. In this case, $n = 3N\ell \times L$, there are L training sequences and each sequence contains $3N\ell$ time frames. The Laplacian pyramid representation is calculated iteratively as follows:

$$\begin{aligned} \boldsymbol{s}_0(t) &= \sum_{j=1}^{n} \boldsymbol{K}_{\mathbf{0}}(t,j)\Gamma_{\mathrm{j}}^{\mathrm{train}} \qquad (4.29) \\ \boldsymbol{s}_{\mathrm{v}}(t) &= \sum_{j=1}^{n} \boldsymbol{K}_{\mathrm{v}}(t,j)\boldsymbol{d}_{\mathrm{v}}(j). \qquad (4.30) \end{aligned}$$

The differences are given by:

$$\boldsymbol{d}_{\mathrm{v}} = \boldsymbol{\Gamma}^{\mathrm{train}} - \sum_{i=0}^{v-1} \boldsymbol{s}_{\mathrm{i}}. \qquad (4.31)$$

The iterations in (4.30) are stopped when $\|\boldsymbol{d}_v\|$ is smaller than a certain threshold. At this point, the training phase is completed. The main parameters which are necessary for the testing phase are the training LRF

and the differences $\boldsymbol{d}_1, .., \boldsymbol{d}_{\mathrm{v}}$. An example of likelihood ratio function represented by the Laplacian pyramid representation is shown in Fig. 4.1.

4.4.3 Testing Phase

During testing, our goal is to decide whether a given unlabeled frame contains speech or not. Mousazadeh and Cohen [19] applied an eigenvector extension and made a speech activity decision by the following likelihood ratio test:

$$\Gamma_{\mathrm{t}}^{\mathrm{test}} = \frac{f(\tilde{\boldsymbol{U}}(t,:); \mathcal{H}_1)}{f(\tilde{\boldsymbol{U}}(t,:); \mathcal{H}_0)} \tag{4.32}$$

where $\tilde{\boldsymbol{U}}(t,:)$ is the extended eigenvectors matrix. We suggest to use Laplacian pyramid algorithm for extending the likelihood ratio function to unlabeled frames.

Let $z(n)$ be the noisy tested signal measured by a microphone. Let $\boldsymbol{Z}(:,t)$ be the feature vector extracted from the t-th unlabeled frame of $z(n)$. For each resolution level v, the similarity matrix between the new data and labeled data is obtained as follows:

$$\boldsymbol{W}_{\theta}^{\mathrm{v}}(i,j) = \exp\left(\sum_{p=-P}^{P} -\alpha_{\mathrm{p}} \boldsymbol{Q}^{\mathrm{v}}(i+p, j+p)\right) \tag{4.33}$$

$$\begin{aligned} \boldsymbol{Q}^{\mathrm{v}}(i,j) = {} & 2^v \| \boldsymbol{Y}_{\mathrm{m}}(:,i)\,(1-\exp(-\Lambda_{\mathrm{i}}/\epsilon)) - \\ & \boldsymbol{Z}_{\mathrm{m}}(:,j)\,(1-\exp(-\Lambda_{\mathrm{j}}/\epsilon))\,\|_2^2, \end{aligned} \tag{4.34}$$

and the likelihood function of unlabeled frames in different resolutions is

given by:

$$\boldsymbol{s}_0^{\mathrm{un}}(t) = \sum_{j=1}^{n} \boldsymbol{K_0}(t,j)\Gamma_{\mathrm{j}}^{\mathrm{train}} \tag{4.35}$$

$$\boldsymbol{s}_{\mathrm{v}}^{\mathrm{un}}(t) = \sum_{j=1}^{n} \boldsymbol{K}_{\mathrm{v}}(t,j)\boldsymbol{d}_{\mathrm{v}}(j). \tag{4.36}$$

when the smoothing operator is calculated as in (4.28).

Finally, the likelihood ratio of the t-th frame is given by:

$$\Gamma_{\mathrm{t}}^{\mathrm{test}} = \sum_{k=0}^{v-1} \boldsymbol{s}_{\mathrm{k}}^{\mathrm{un}}(t). \tag{4.37}$$

Frames containing speech are usually followed by a frame which also contains speech while the transient signals usually last only for few time frames. Therefore, the decision rule for an unlabeled frame can be calculated by:

$$VAD = \sum_{j=-J}^{J} \Gamma_{\mathrm{t+j}}^{\mathrm{test}} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} T_h \qquad t = 1, 2, \cdots, T \tag{4.38}$$

where T_h is the threshold which controls the tradeoff between probability of detection and false alarm. Increasing (decreasing) this parameter leads to decrease (increase) of both the probability of false alarm and the probability of detection.

4.5 Experimental Results

In this section, we examine the performance of the proposed method using several simulations. We compare the performance of our method to those of conventional statistical model-based methods presented in [16,20,50,52],

and the VAD proposed by Mousazadeh and Cohen [19] which is based on spectral clustering.

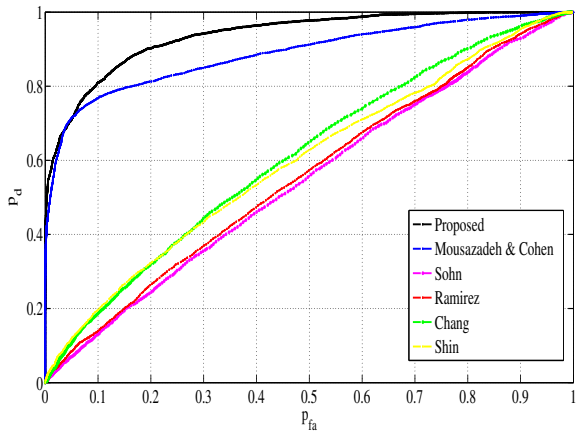
4.5.1 Experimental Setup

We perform our simulation under different types of stationary and transient noises, and at different SNR levels. We use different data during the training and the testing phases. We challenge the algorithm by taking 5 training sequences and 30 testing sequences at each simulation. Speech signals are taken from the TIMIT database [45], and transient noise signals are taken from [47]. The sampling frequency is 16kHz. We use STFT with frame length of 512 samples, 50% overlap and a hamming window. We compute the MFCCs in $K_m = 24$ Mel frequency bands and the *a-priori* threshold which is used in the Laplacian pyramid algorithm is set to 10^{-3}. In order to compare our method to a conventional statistical based method, we introduce two different kinds of false alarm probabilities. The first one is denoted by P_{fa} and is defined as the probability that a speech free frame is detected as a speech frame. The second type is denoted by P_{fatr} and is defined as the probability that a frame consisting of stationary and transient noise is detected as a speech frame. We need these two concepts to show the advantage of the proposed method over conventional statistical model-based methods. The number of frames that contain transient noise and are mostly detected as speech in statistical model-based methods, is relatively low and therefore do not significantly affect the probability of false alarms. Regard the kernel parameters, in all simulations we set the parameter vector to $\boldsymbol{\theta} = .001 \times [300\ 0.4\ 0.75\ 1\ 0.75\ 0.4]$.

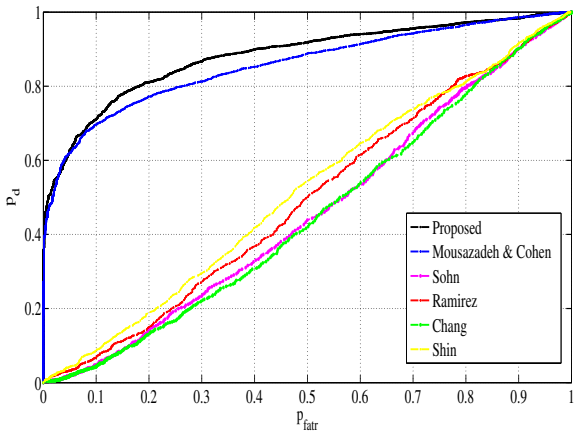
4.5.2 Performance Evaluation

The simulation results are presented in Fig. 4.5. We have compared the proposed algorithm to state-of-the-art VAD algorithms which are based on log of the likelihood ratio test. The difference between the compared algorithms is the statistical model that is utilized for the estimation of the likelihood ratio. Sohn et al. [20] utilize the Gaussian model in order to estimate the likelihood ratio. Chang et al. [16] and Shin et al. [50] utilize the Laplacian and Gamma model, respectively. Remirez et al. [52] use the concept of multiple observation probability ratio test, and calculate the likelihood ratio in past and future frames. Similar to Sohn, they also assumed Gaussian model for speech and noise signals. A common assumption for those compared algorithms is that noise is slowly varying with respect to speech. This assumption is not suitable in case of transient noise. In addition, modeling transients is not reasonable as modeling stationary noise. As consequence, these algorithms fail in case of very noisy environments which are characterized by transient noise. Although different statistical model-based methods have different performance in different situations, the proposed method has superior performance in all simulations over the compared methods, particulary for low false alarm rates. However, the main drawback of the proposed algorithm is the computational load which is relatively higher than statistical-based VADs. For example, the processing time of a 6 minutes conversation is approximately one minute.

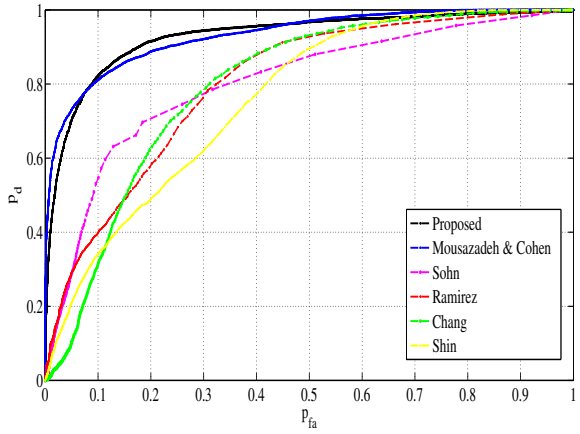
When comparing to the method presented by Mousazadeh and Cohen [19], our method demonstrates an improved performance, especially



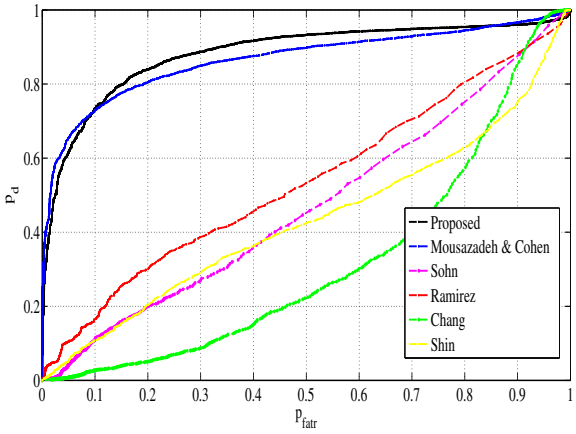
(a)



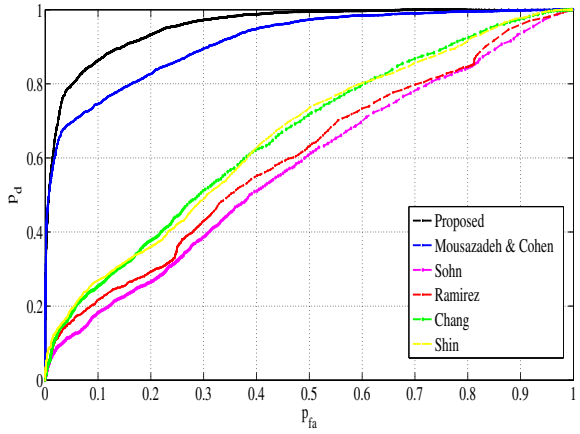
(b)



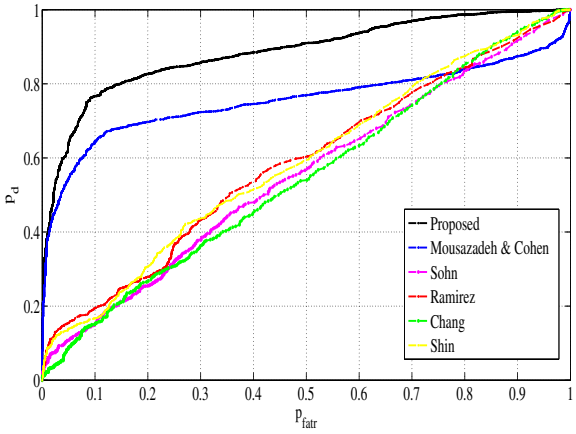
(c)



(d)



(e)



(f)

Figure 4.2: P_{d} versus P_{fa} (left column), P_{d} versus P_{fatr} (right column) for different noise environments. Figures (a)-(b): SNR = 0dB, stationary noise - white Gaussian, transient noise - keyboard stroke. Figures (c)-(d): SNR = 10dB, stationary noise - babble noise, transient noise - keyboard stroke. Figures (e)-(f): SNR = 20dB, stationary noise - colored noise, transient noise - door knocks.

in cases of small number of training sequences and at low SNR levels.
The main difference between the proposed algorithm and the algorithm introduced in [19] is the implemented function extension method. While Mousazadeh and Cohen proposed to extend the eigenvectors representation, we extended the likelihood ratios. They estimated the eigenvectors matrix of the testing sequences ($\boldsymbol{U}^{test}$) by a linear combination of the training matrix $\boldsymbol{U}^{train}$. The combination depends on a matrix marked as $\boldsymbol{B}$ that defines the similarity matrix between the training and the testing feature vectors. The Laplacian pyramid representation which is used in our VAD, can also be considered as a linear combination that depends on a similarity matrix between the training and testing data points. However, the Laplacian pyramid algorithm uses directly the training likelihood ratio function. Since Mousazadeh and Cohen rely only on the eigenvectors of the training sequences, a significant performance degradation may occur in case of underlying incorrect structure. Therefore, the proposed algorithm demonstrates improved results, particulary in cases of small training files.

4.6 Conclusions

We have presented a supervised algorithm for voice activity detection. The proposed method is based on Laplacian pyramid algorithm which is utilized as a function extension tool. The likelihood ratio function was extended to new unlabeled points and then, compared to a threshold in order to decide whether each frame is comprised of speech or non-speech signals. Our main concern was dealing with low SNR transient noise conditions, which are difficult to handle. Therefore, we utilized features which are suitable for

separating between speech and non-speech frames in noisy environments. Then, we calculated the likelihood ratios using spectral clustering. Finally, we used Laplacian pyramid algorithm for extending the likelihood ratio function to a new set of points. Simulation results demonstrate the improved performance of the proposed method and particularly its advantage in treating transient noise using only few training sequences.

Chapter 5

Research Summary and Future Directions

5.1 Research Summary

In this research, we have addressed the problem of speaker diarization, focusing on short speech utterances (rapid speaker change point) and noisy environments include stationary and transient noises. We presented a novel method which is based on GMM supervectors and spectral clustering. The first stage of the algorithm was segmentation into isolated segments which are comprised of one active speaker each. We showed that segmentation of short utterances by conventional BIC-based speaker change point detector fails, and utilizing VAD for this purpose is preferred. Therefore, we implemented a VAD algorithm which handles with noisy conditions. The proposed VAD is based on graph embedding method (spectral clustering) and Laplacian pyramid algorithm, which constitutes as a function extension tool. Finally, by comparing the likelihood ratios of unlabeled frames to a given threshold, we decided whether the frame consists of speech or not. The usage of VAD enables fine segmentation which is necessary for

dealing with diarization of short utterances.

Each segment of speech (i.e., an utterance) was represented by a GMM mean supervector. We used also its first and second derivatives, which show that there is a significant information in variations over time. Finally, we applied spectral clustering on the obtained set of supervectors. The spectral clustering enables to underly the structure of the data and to represent the supervectors by low dimensional feature vectors, without assuming in advance linearity or any other type of data structure. Those features describe a low-dimensional manifold which can be represented as a graph that specifies neighbors points on the manifold. We have also shown that there is a strong connection between the eigenvalues spectrum and the number of clusters. The estimated number of clusters is obtained by searching the largest gap between consecutive eigenvalues. This number is used as an input to the k-means algorithm. In addition, we compared between two types of metric distance: KL divergence and cosine metric. The cosine metric has two main advantages. First, it neglects the magnitude which assumed to be uninformative. Second, it enables to combine the first and second derivatives of the GMM supervector. The simulation results demonstrate these advantages over the KL divergence.

To conclude, the main research contributions are as follows. First, development of a unique VAD algorithm, which better handles noisy environments compared to with state-of-the-art VAD algorithms. This VAD also enables a fine segmentation which is necessary when dealing with rapid speaker change point points during a conversation. Second, the supervector representation followed by spectral clustering enhances speaker discrimina-

tion even in short utterances cases and enables a quality estimation of the number of involved speakers. In addition, the spectral clustering improves the robustness of the speaker diarization to noise.

5.2 Future Research Directions

The presented research in this thesis, opens some directions that can be researched in a future work.

1. In this dissertation, we assumed no speaker overlap during the conversation, however speaker overlap is a common phenomena which takes a place in many spontaneous and nature conversations. It includes two main sub-tasks: Detecting the overlap speech segments and deciding the dominant speaker as well as the overlapped speakers. Focusing on detecting the overlapped segments, the most important stage is feature extraction. Besides the familiar features such as MFCCs and their derivatives, LPC, etc, a unique feature must be developed. It is worth to research a type of entropy measure as a feature. In non-overlapped speech segments we expect a lower entropy than in overlapped speech segments. A different research direction may be a supervised learning algorithm based on SVM, which is trained on database includes overlapped and non-overlapped speech segments. The SVM is a suitable choice for a binary problem (Overlapped or non-overlapped segment).

2. Diarization of movies has few references in the literature and therefore can be considered as a new research direction. The main difficulties in movies diarization are: High number of speakers, speaker overlaps,

background music, rapid changes etc. In this aspect, we suggest to research a different view of diarization system which is an audio-visual diarization. The audio-visual diarization system fuses audio diarization and face recognition, promising improved performance.

3. In this dissertation, we used spectral clustering in order to reveal the underlying structure of the audio signal. A Significant phase in spectral clustering is choosing the kernel which is necessary for calculating the weight and Laplacian matrices. We chose the Gaussian kernel, however a worthy research direction may be a deep investigation of other similarity matrices as well as a development of a specific weight matrix. One suggestion is extracting a supervector which consists of main speaker features such as MFCCs, pitch frequency, direction of arrival (DOA) (in case of room meetings), etc. Then, implementing a weight matrix such that each feature is enhanced separately and the combination of all involved features leads to an improved speaker discrimination. For example, multiplication of Gaussian kernels such that each exponent has different weight.

4. We introduced a speaker diarization system based on supervectors and spectral clustering. The output of this algorithm, which is speaker clusters, can constitutes as an input to speaker identification system. Assuming a training database, a supervector can be extracted for each speaker. One research direction may be re-modeling each obtained cluster by a supervector and find the most similar labelled supervector. For example, applying spectral clustering and set number of clusters

to K, supposing the result will be $K-1$ clusters consist one data point each and one cluster include two data points.

In addition, short speaker utterances identification, particulary under noisy environments are still open issues. Therefore, a different research direction is an independent speaker identification system based on the proposed method.

5. Meeting room configuration and microphone placement affect the recording quality, including background noise and reverberation. A reverberation is created when a sound or signal is reflected causing a large number of reflections to build up and then decay as the sound is absorbed by the surfaces of objects in the space. The reverberation may cause mistakes in detecting the speech segment and tagging segments of reverberation as a silence or as an another speaker. Therefore, it is a new research direction which requires deep investigation including evaluation of the system performance under reverberation conditions and an improved system implementation.

Bibliography

[1] P. Kenny, D. Reynolds, and F. Castaldo, "Diarization of telephone conversations using factor analysis," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 4, no. 6, pp. 1059–1070, 2010.

[2] D. Charlet, C. Barras, and J.-S. Lienard, "Impact of overlapping speech detection on speaker diarization for broadcast news and debates," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on.* IEEE, 2013, pp. 7707–7711.

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

[4] M. H. Moattar and M. M. Homayounpour, "A review on speaker diarization systems and approaches," *Speech Communication*, vol. 54, no. 10, pp. 1065–1103, 2012.

[5] S. E. Tranter and D. A. Reynolds, "An overview of automatic speaker diarization systems," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 5, pp. 1557–1565, 2006.

[6] A. Jansen and P. Niyogi, "A geometric perspective on speech sounds," *University of Chicago, Tech. Rep*, 2005.

[7] X. Anguera Miro, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker diarization: A review of recent research," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 2, pp. 356–370, 2012.

[8] X. Anguera, C. Wooters, and J. M. Pardo, "Robust speaker diarization for meetings: Icsi rt06s meetings evaluation system," in *Machine Learning for Multimodal Interaction.* Springer, 2006, pp. 346–358.

[9] T. Nguyen, H. Sun, S. Zhao, S. Khine, H. Tran, T. Ma, B. Ma, E. Chng, and H. Li, "The iir-ntu speaker diarization systems for rt 2009," in *RT09, NIST Rich Transcription Workshop, May 28-29, 2009, Melbourne, Florida, USA*, vol. 14, 2009, pp. 17–40.

[10] S. Bozonnet, N. W. Evans, and C. Fredouille, "The lia-eurecom rt'09 speaker diarization system: enhancements in speaker modelling and cluster purification," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on.* IEEE, 2010, pp. 4958–4961.

[11] G. Friedland, O. Vinyals, Y. Huang, and C. Muller, "Prosodic and other long-term features for speaker diarization," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 17, no. 5, pp. 985–993, 2009.

[12] V. Dellwo, M. Huckvale, and M. Ashby, "How is individuality expressed in voice? an introduction to speech production and description for speaker classification," in *Speaker Classification I.* Springer, 2007, pp. 1–20.

[13] H. Tang, S. M. Chu, M. Hasegawa-Johnson, and T. S. Huang, "Partially supervised speaker clustering," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 5, pp. 959–971, 2012.

[14] S. Shum, N. Dehak, and J. Glass, "On the use of spectral and iterative methods for speaker diarization," *System*, vol. 1, no. w2, p. 2, 2012.

[15] S. Gazor and W. Zhang, "A soft voice activity detector based on a laplacian gaussian model," *IEEE Trans. Speech and Audio Processing*, vol. 11, pp. 498–505, 2003.

[16] J.-H. Chang and N. S. Kim, "Voice activity detection based on complex laplacian model," *Electronics Letters*, vol. 39, no. 7, pp. 632–634, 2003.

[17] A. Davis, S. Nordholm, and R. Togneri, "Statistical voice activity detection using low-variance spectrum estimation and an adaptive threshold," *IEEE Trans. Audio, Speech and Language Processing*, vol. 14, no. 2, pp. 412–424, 2006.

[18] K. Ishizuka, T. Nakatani, M. Fujimoto, and N. Miyazaki, "Noise robust voice activity detection based on periodic to aperiodic component ratio," *Speech communication*, vol. 52, no. 1, pp. 41–60, 2010.

[19] S. Mousazadeh and I. Cohen, "Voice activity detection in presence of transient noise using spectral clustering," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, no. 6, pp. 1261–1271, 2013.

[20] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection," *Signal Processing Letters, IEEE*, vol. 6, no. 1, pp. 1–3, 1999.

[21] N. Spingarn, S. Mousazadeh, and I. Cohen, "Voice activity detection in transient noise environment using laplacian pyramid algorithm," in *Acoustic Signal Enhancement (IWAENC), 2014 14th International Workshop on Acoustic Signal Enhancement*. IEEE, 2014, pp. 238–242.

[22] S. Chen and P. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the bayesian information criterion," in *Proc. DARPA Broadcast News Transcription and Understanding Workshop*. Virginia, USA, 1998, p. 8.

[23] H. Gish, M.-H. Siu, and R. Rohlicek, "Segregation of speakers for speech recognition and speaker identification," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on*. IEEE, 1991, pp. 873–876.

[24] F. R. Bach and M. I. Jordan, "Learning spectral clustering, with application to speech separation," *The Journal of Machine Learning Research*, vol. 7, pp. 1963–2001, 2006.

[25] H. Ning, M. Liu, H. Tang, and T. S. Huang, "A spectral clustering approach to speaker diarization." in *INTERSPEECH*, 2006.

[26] X. Niyogi, "Locality preserving projections," in *Neural information processing systems*, vol. 16, 2004, p. 153.

[27] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[28] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.

[29] F. R. Chung, *Spectral graph theory*. American Mathematical Soc., 1997, vol. 92.

[30] A. Y. Ng, M. I. Jordan, Y. Weiss *et al.*, "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 2, pp. 849–856, 2002.

[31] X. He, D. Cai, S. Yan, and H.-J. Zhang, "Neighborhood preserving embedding," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2. IEEE, 2005, pp. 1208–1213.

[32] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using gmm supervectors for speaker verification," *Signal Processing Letters, IEEE*, vol. 13, no. 5, pp. 308–311, 2006.

[33] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: From features to supervectors," *Speech communication*, vol. 52, no. 1, pp. 12–40, 2010.

[34] P. Kenny and P. Dumouchel, "Experiments in speaker verification using factor analysis likelihood ratios," in *ODYSSEY04-The Speaker and Language Recognition Workshop*, 2004.

[35] L. Muda, M. Begam, and I. Elamvazuthi, "Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques," *arXiv preprint arXiv:1003.4083*, 2010.

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

[37] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *Speech and audio processing, ieee transactions on*, vol. 2, no. 2, pp. 291–298, 1994.

[38] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Advances in neural information processing systems*, 2004, pp. 1601–1608.

[39] M. Ben, M. Betser, F. Bimbot, and G. Gravier, "Speaker diarization using bottom-up clustering based on a parameter-derived distance between adapted gmms," in *Proc. ICSLP*, vol. 2004, 2004.

[40] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788–798, 2011.

[41] N. Dehak, R. Dehak, J. R. Glass, D. A. Reynolds, and P. Kenny, "Cosine similarity scoring without score normalization techniques." in *Odyssey*, 2010, p. 15.

[42] M. N. Do, "Fast approximation of kullback-leibler distance for dependence trees and hidden markov models," *Signal Processing Letters, IEEE*, vol. 10, no. 4, pp. 115–118, 2003.

[43] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[44] D. Moraru, L. Besacier, and E. Castelli, "Using a priori information for speaker diarization," in *ODYSSEY04-The Speaker and Language Recognition Workshop*, 2004.

[45] J. S. Garofolo, "Getting started with the DARPA TIMIT CD-ROM: An acoustic-phonetic continous speech database," National Inst. of Standards and Technology (NIST), Gaithersburg, MD, Feb 1993.

[46] "[online]. available: http://www.http://festvox.org/."

[47] "[online]. available: http://www.freesound.org."

[48] NIST, "Diarization Error Rate (DER) Scoring Code," 2006. [Online]. Available: www.nist.gov/speech/tests/rt/2006-spring/code/md-evalv21.pl

[49] F. Valente and C. Wellekens, "Variational bayesian speaker clustering," in *ODYSSEY04-The Speaker and Language Recognition Workshop*, 2004.

[50] J. W. Shin, J. H. Chang, H. S. Yun, and N. S. Kim, "Voice activity detection based on generalized gamma distribution," *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 1, pp. I781–I784, 2005.

[51] S. Mousazadeh and I. Cohen, "AR-GARCH in presence of noise: Parameter estimation and its application to voice activity detection," *IEEE Trans. Audio, Speech and Language Processing*, vol. 19, no. 4, pp. 916–926, 2011.

[52] J. Ramirez and J. C. Segura, "Statistical voice activity detection using a multiple observation likelihood ratio test," *IEEE Signal Process. Lett.*, vol. 12, pp. 689–692, 2005.

[53] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 32, p. 443445, 1985.

[54] I. Cohen, "Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging," *Speech and Audio Processing, IEEE Transactions on*, vol. 11, no. 5, pp. 466–475, 2003.

[55] P. J. Burt and E. H. Adelson, "The laplacian pyramid as a compact image code," *Communications, IEEE Transactions on*, vol. 31, no. 4, pp. 532–540, 1983.

[56] N.Rabin and R. Coifman., "Heterogeneous datasets representation and learning using diffusion maps and laplacian pyramids," in *Proc. 12th SIAM International Conference on Data Mining., Speech Date Mining .,(SDM)*, 2012.

[57] G. H. Golub and C. F. V. Loan, *Matrix Computations*. Johns Hopkins University Press, 1996.

תיוג דוברים באמצעות סופר - וקטורים
ואלגוריתמים מתקדמים
להורדת מימדיות

נורית שפינגרן

תיוג דוברים באמצעות סופר - וקטורים ואלגוריתמים מתקדמים להורדת מימדיות

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר
מגיסטר

נורית שפינגרן

הוגש לסנט הטכניון - מכון טכנולוגי לישראל

טבת תשע״ו ,חיפה , ינואר 2016

המחקר נעשה בהנחיית פרופ׳ ישראל כהן
מהפקולטה להנדסת חשמל בטכניון

תודות

ברצוני להביע את הערכת ותודתי הרבה לפרופ׳ ישראל כהן עבור ההנחייה המקצועית והמסורה, על העידוד למצויינות והתרומה הרבה והמשמעותית להתפתחותי האישית.

כמו כן, תודות רבות לצוות מעבדת SIPL על העזרה והתמיכה במהלך המחקר. סאמאן מוסזאדה והדס בניסתי, תודה על העצות המועילות, העידוד והחברות הקרובה.

ברצוני להודות למשפחתי היקרה, להורי, רפאל ז״ל ונעמי, ולבעלי דוד על אהבה אינסופית, עידוד ותמיכה לאורך כל הדרך.

אני מודה לטכניון ולקרן הלאומית למדע (מענק מס׳ 1130/11) על התמיכה הכספית הנדיבה בהשתלמותי.

תקציר

תיוג דוברים בשיחה (Speaker Diarization) הינו תחום מרכזי בעיבוד אותות דיבור הזוכה למאמץ מחקרי משמעותי בעשור האחרון. אף על פי כן, קיימות מספר סוגיות מאתגרות אשר טרם נמצא להן מענה מספק ודורשות מחקר נוסף. עבודה זו עוסקת בפיתוח אלגוריתם לתיוג דוברים המתמודד עם שני קשיים מרכזיים בתחום – חילוף דוברים מהיר וסביבה רועשת.

תחילה, האלגוריתם המוצע עוסק בבעית זיהוי דיבור (VAD-Voice Activity Detection) בנוכחות רעשי רקע סטציונריים וטרנזינטיים. אלגוריתם לזיהוי דיבור מהווה שלב ראשוני במערכות עיבוד אותות רבות כמו גם במערכות אשר משמשות לתיוג, זיהוי או אימות דוברים (Speaker Identification\Verification). במסגרת זאת, פותח אלגוריתם למידה לזיהוי דיבור המבוסס על שיטת ה-Laplacian Pyramid. פלט האלגוריתם מורכב משני אשכולות (Clusters), כאשר האחד מייצג מסגרות דיבור (speech frames) והשני מסגרות של היעדר דיבור. אלגוריתם ה-VAD המוצע מבוסס על מבחן יחס הסבירויות (LRT - Likelihood Ratio Test), אשר מטרתו להכריע האם מסגרת האות הנתונה משתייכת לאשכול הדיבור או לאשכול היעדר הדיבור. במידה והערך המתקבל גדול מסף מסויים אזי מדובר במסגרת המכילה דיבור ולהיפך.

אלגוריתם ה-VAD המוצע מתבצע באופן הבא. בשלב האימון, אנו נעזרים בקבצי דיבור המכילים מספר רב של רעשים ומחשבים את פונקצית יחס הסבירויות. פונקציה זו מיוצגת מחדש ע״י אלגוריתם ה-Laplacian Pyramid ומתקבלים מספר פרמטרים אשר מועברים לשלב הבוחן בו אנו מרחיבים את הפונקציה למסגרות חדשות שלא נראו בעבר. תהליך זה נקרא גם ״הרחבת הפונקציה״ (Function Extension). על מנת לייצר את פונקצית יחס הסבירויות של סט האימון אנו נעזרים בשיטת ה-Spectral Clustering. תחילה, אנו מחשבים את הוקטורים העצמיים של מטריצת הלפלאסיאן, ולאחר מכן ממדלים את רכיביהם על פי השייכות למסגרות דיבור או היעדר דיבור. המידול נעשה על ידי מודל GMM (Gaussian Mixture Model) כך שלכל אשכול (דיבור או היעדר דיבור) קיים מודל GMM אשר מייצג אותו.

ההתמודדות עם רעשי הרקע החריפים באה לידי ביטוי במספר רבדים באלגוריתם זה. הראשון מבוסס על הוצאת מאפיינים ייחודים המוכחים כיעילים בתנאי רעש. הרובד השני נשען על שימוש בקונספט ה-Spectral clustering. שיטה זו נמנית על שיטות רבות של Graph embedding אשר מאפשרות לייצג את המימדים האינפורמטיביים ביותר של המידע, ובין היתר להמנע מייצוג של הרעש. הרובד השלישי מורכב מייצוג מחדש של המידע על ידי אלגוריתם ה-Laplacian Pyramid אשר מאפשר הרחבה מוצלחת של הפונקציה.

תוצאות הסימולציה מדגימות את יתרונה של השיטה המוצעת בהשוואה לאלגוריתמים קונבנציונליים המבוססים על מודלים סטטיסטיים. בניגוד לאלגוריתמים הקונבנציונאליים, האלגוריתם המוצע אינו מניח שינוי איטי של הרעש ביחס לדיבור או לחלופין מודל סטטיסטי של מאפייני הדיבור עצמם, אלא מבוסס על מידול של הממדים האינפורמטיביים ביותר של המידע. מידול זה מביא לידי דחייה של אלמנטי הרעש והדגשה של הממדים העיקריים המראים על קיומו

של דיבור. כתוצאה מכך, מתקבל בידול איכותי בין מסגרות הדיבור ומסגרות היעדר הדיבור. פיזור של נקודות המידע על פני הממדים העיקריים של המידע מראה על כך באופן מובהק. כמו כן, ניכר מהתוצאות כי שימוש באלגוריתם ה-Laplacian Pyramid שבאמצעותו מחושבים ערכי יחס הסבירויות של מסגרות הבוחן, מאפשר שליטה ב-Trade off שבין ייצוג מדוייק של הערכים המתקבלים לבין התאמת יתר (Over-fitting) של מידע הבוחן באמצעות בחירת מספר רמות הייצוג.

בעיית תיוג הדוברים מורכבת משתי תתי-בעיות: סגמנטציה ואישכול (Clustering). מטרת הסגמנציה הינה לחלק את שיחת הבוחן למקטעים כך שכל מקטע מורכב מדובר יחיד בלבד. ה-VAD שהוצע לעיל מאפשר לנו לבצע סגמנטציה עדינה מאוד המתאימה לשיחה המאופיינת בחילוף דוברים מהיר. כמו כן, הוא מונע עיבוד של חלקי הרעש כחלק מהדיבור אשר עלול לפגוע באופן משמעותי בביצועים. השיטות הקונבנציונאליות למציאת שינוי דובר (סגמנטציה) מבוססות על קריטריון בייסיאני (BIC-Bayesian Criterion Information), אשר חסרונו מתגלה במקרה של חילוף דוברים מהיר. פלט האלגוריתם לרוב מניב סגמנטים אשר מורכבים משני דוברים, אף על פי שבפועל היו אמורים להכיל דובר אחד בלבד. כתוצאה מכך, עיבוד ההמשך של האלגוריתם הכולל אישכול (Clustering) נפגע משמעותית.

לאחר חלוקת השיחה לסגמנטים מבודדים, אנו ממדלים כל סגמנט באמצעות סופר-וקטור. מדובר בשיטה לייצוג של מקטעי דיבור על ידי וקטור ארוך המורכב משרשור של וקטורי תוחלות ה-GMM. ייצוג זה מאפשר הבחנה חדה יותר, ביחס לשיטות אחרות, בין הדוברים המעורבים בשיחה. מציאת הסופר וקטורים נעשית על ידי אדפטציה מבוססת מקסימום א-פריורי (Maximum A-priori - MAP) של מודל "הדובר הכללי" (UBM-Universal Background Model) לקטע הדיבור הנוכחי. השוני בין המודל הכללי לוקטור המתקבל מאפשר הבחנה בין דוברים שונים. כמו כן, בשיטה המוצעת אנו מחשבים גם את נגזרות הסופר-וקטורים אשר מחדדות את ההבחנה בין הדוברים השונים שכן הן מייצגות שינוי לאורך זמן המדגיש את השוני בין סופר וקטורים השייכים לדוברים שונים. יתרה מזאת, ייצוג הסגמנט באמצעות וקטור יחיד מהווה פלטפורמה נוחה לשימוש באלגוריתמים להורדת ממדיות שהינם יעילים מאוד בתנאי רעש.

הסופר וקטורים משמשים כקלט לאלגוריתם ה-Spectral Clustering. לטובת חישוב מטריצת הדמיון, שהינה שלב מרכזי, נשתמש בגרעין הגאוסי כאשר המרחק בין שני סגמנטים יחושב על ידי מרחק קוסינוסי או דיברגנס KL (Kullback–Leibler divergence). למרחק הקוסינוסי יתרון על פני מרחקים אחרים, שכן בחישוב זה אין התייחסות לאמפליטודה, אשר אינה מכילה מידע קריטי ואף יכולה להיפגע משמעותית לאור מאפייני ערוץ או תווך שונים. לפיכך, שימוש במרחק הקוסינוסי יכול לסייע בשיפור רובוסטיות המערכת כנגד רעש.

קושי ידוע בבעית תיוג דוברים הינו מציאת מספר הדוברים בבעיה. אלגוריתמים רבים מניחים כי מספר הדוברים בשיחה ידוע מראש ובכך מקלים על הבעיה. באלגוריתם המוצע, אנו מניחים כי מספר הדוברים אינו ידוע. אנו מתמודדים עם הבעיה באמצעות בחינת התנהגות הערכים העצמיים של מטריצת הלפלאסיאן. מהניתוח עולה כי מספר הערכים העצמיים הגבוהים קורלטיבי למספר האשכולות ולמעשה מעיד על מספר הדוברים בבעיה.

האלגוריתם המוצע נבחן על פני שיחות מוקלטות וסינתתיות רבות ומאתגרות במיוחד, כאשר כולן מאופיינות על ידי חילוף דוברים מהיר (כל קטע דיבור אורך בממוצע כ-3 שניות) ורעשי רקע שונים. ביצועי המערכת נאמדים ע״י שני מדדים שונים: מדד ה-DER (Diarization Error Rate) ומדד ה-ACP (Average Clustering Purity). תוצאות הסימולציה מראות על תוצאות מרשימות במיוחד ביחס למערכות תיוג דוברים מתחרות.

הנה כי כן, שיטת ה-Spectral Clustering מאפשרת למצוא את המימדים האינפורמטיביים ביותר של המידע ולייצג באופן יעיל כל סגמנט ע״י וקטור מממד נמוך, כך שמתקבלת הפרדה ברורה בין דוברים שונים. השיטה אינה מוגבלת למציאת מניפולד ליניארי ובה בעת מחוייבת לשמור על יחסי קרבה בין נקודות שכנות במובן של מטריקה נבחרת (נקודות אשר היו קרובות במימד הגבוה ישארו קרובות במימד הנמוך). בעבודה זו, אנו עושים שימוש ביתרונות אלה על מנת לבצע לימוד נבון של המידע והפרדה איכותית בין סגמנטי דיבור אשר אינם שייכים לאותו הדובר.