Speech and Audio Processing Based on Harmonic Analysis and Statistical Models

Saman Mousazadeh

Speech and Audio Processing Based on Harmonic Analysis and Statistical Models

Research Thesis

As Partial Fulfillment of the Requirements for
the Degree Doctor of Philosophy

Saman Mousazadeh

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

Adar I 5776 Haifa February 2016

Acknowledgement

I wish to express my deep gratitude and appreciation to Prof. Israel Cohen for his guidance and dedicated supervision. I greatly appreciate his professional support, encouragement to perfection and contribution to my personal development which goes way beyond the scope of this work. I would also like to thank Prof. David Malah for his valuable inputs and helpful advice. I would also like to thank Nimrod Peleg and Avi Rosen for their valuable support.

I would like to thank my friends and colleagues in the Technion. Special thanks to Idan Ram, Nurit Spingarn, Ronen Talmon, Tomer Peleg and David Dov for their support and friendship, for their helpful comments, and for making my stay in the Technion fun and memorable.

I would like to thank Elahe. Her support, encouragement, quiet patience and unwavering love were undeniably the bedrock upon which a part of my life have been built. Her tolerance of my occasional vulgar moods is a testament in itself of her unyielding devotion and love.

Most importantly, I would like to express my deep gratitude to my dear family - my parents Benyamin and Ester and to my sisters Dorna and Golnoush, for their love, encouragement and support.

Last but not least, thanks to my beloved daughter, Neta, whose smiles are the most beautiful thing I have ever seen.

The Generous Financial Help Of The Technion, The Irwin And Joan Jacobs Fellowship, Gutwirth Fellowship, Ollendorff Fellowship And The Israel Science Foundation (Grant no. 1130/11) Is Gratefully Acknowledged.

List of Publications

1. S. Mousazadeh and I. Cohen ”Voice Activity Detection in Presence of Transient Noise Using Spectral Clustering” IEEE Trans. Audio, Speech and Language Processing, Vol. 21, Number 6, June 2013, pp. 1261-1271.

2. S. Mousazadeh and I. Cohen ”Embedding and Function Extension on Directed Graph” Submitted to Signal Processing.

3. S. Mousazadeh and I. Cohen ”Out-of-Sample Extension of Band-Limited Functions on Homogenous Manifolds Using Diffusion Maps” Signal Processing, Vol. 108, March 2015, pp 521529.

4. S. Mousazadeh and I. Cohen ”Simultaneous Parameter Estimation and State Smoothing of Complex GARCH Process in the Presence of Additive Noise” Signal Processing, Vol. 90, Issue 11, Nov. 2010, pp. 2947-2953.

5. 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, Issue 4, May 2011, pp. 916-926.

6. S. Mousazadeh and I. Cohen ”Anomaly Detection in Sonar Images Based on Wavelet Domain Noncausal AR-ARCH Random Field Modeling” Proc. 26th IEEE Convention of Electrical and Electronics Engineers in Israel, IEEEI-2010, Eilat, Israel, 17-20 November 2010.

7. S. Mousazadeh and I. Cohen ”Two dimensional noncausal AR-ARCH model: stationary conditions, parameter estimation and its application to anomaly detection” Signal Processing, Vol. 98, May 2014, pp 322336.

8. S. Mousazadeh and I. Cohen ”ARCH and GARCH Parameter Estimation in Presence of Additive Noise Using Particle Methods” Proc. 38th IEEE Internat. Conf. Acoust. Speech Signal Process., ICASSP-2013, Vancouver, Canada, May 26-31, 2013, pp. 6279-6282.

Contents

1 Introduction **7**

1.1 Geometric Tools for Data Analysis 8

1.1.1 A Brief Introduction to Diffusion Maps 9

1.1.2 Voice Activity Detection in Presence of Transient Noise Using Spectral Clustering 10

1.1.3 Embedding and Function Extension on Directed Graph 13

1.1.4 Out-of-Sample Extension of Band-Limited Functions on Homogenous Manifolds Using Diffusion Maps 15

1.2 Model-based speech signal processing 17

1.2.1 Simultaneous Parameter Estimation and State Smoothing of Complex GARCH Process in the Presence of Additive Noise 19

1.2.2 AR-GARCH in Presence of Noise: Parameter Estimation and its Application to Voice Activity Detection 20

1.2.3 Two Dimensional Noncausal AR-ARCH Model: Stationary Conditions, Parameter Estimation and its Application to Anomaly Detection . 21

1.2.4 ARCH and GARCH Parameter Estimation in Presence of Additive Noise Using Particle Methods 23

1.3 Thesis Structure 24

1.4 List of Publications 26

2 VAD in Presence of Transient Noise **28**

2.1 Introduction 28

2.2 Problem Formulation 31

2.2.1 Feature Selection 31

2.2.2 Clustering Algorithm 32
2.3 Simulation Results and Performance Evaluation 42
2.4 Conclusions 50

3 Embedding on Directed Graph 55
3.1 Introduction 55
3.2 Problem Formulation, Embedding and Function Extension 58
3.2.1 Anisotropic Diffusion Operator 58
3.2.2 Continuous Limit of the Diffusion Operator 60
3.2.3 Manifold Learning, Estimation of Sampling Density and the Vector Field 63
3.2.4 Embedding and Function Extension on Directed Graph 63
3.3 Experimental Results 65
3.4 Conclusions 69
3.5 Appendix A: Proof of Lemma 3.2.1 69
3.6 Appendix B: Proof of Theorem 3.2.3 74

4 Out-of-Sample Extension of Band-Limited ... 82
4.1 Introduction 82
4.2 Problem Formulation 85
4.3 Applications, Simulation Results and Performance Evaluation 89
4.4 Conclusions 95

5 Simultaneous Parameter Estimation ... 101
5.1 Introduction 101
5.2 Problem formulation 102
5.3 ML Estimation of the Parameters 103
5.3.1 Recursive ML 109
5.4 State smoothing 110
5.5 Simulations 111
5.5.1 Computational Complexity 115
5.6 Conclusion 115

6 AR-GARCH in Presence of Noise 117

6.1 AR versus AR-GARCH modeling of speech signals 119

6.2 AR-GARCH parameter estimation 121

6.2.1 Considering the Stationarity Conditions 126

6.3 VAD based on AR-GARCH modeling of speech signals in time domain 128

6.4 Performance Evaluation 130

6.5 Conclusions 138

7 Two Dimensional Noncausal AR-ARCH Model 140

7.1 Introduction 140

7.2 Noncausal ARCH Model 142

7.2.1 Two dimensional Noncausal ARCH Model and its Stationary Conditions 142

7.2.2 Parameter Estimation of Two Dimensional Noncausal ARCH Model . . 146

7.3 Anomaly Detection Based on Noncausal AR-ARCH Model 150

7.4 Performance Evaluation and Simulation Results 154

7.5 Conclusions 158

7.6 Appendix 160

8 ARCH and GARCH Parameter Estimation ... 168

8.1 Introduction 168

8.2 Problem Formulation and the Proposed Method for Parameter Estimation . . . 169

8.3 simulation results 174

8.4 Conclusions 175

9 Research Summary and Future Direction 176

9.1 Research Summary 176

9.2 Future Research Directions 179

Bibliography 181

List of Figures

2.1 A typical training sequence consisting of speech signal and keyboard stoke noise (transient noise) corrupted by babble noise (absolute value of MFCC (right) likelihood ratio (left)). 35

2.2 Scatter plot of the new representation of a typical training data sequence containing speech and door knock noise degraded by colored Gaussian noise (SNR=5dB) obtained by concatenating all training sequences. 39

2.3 Scatter plot of the new representation of a typical training data sequence containing speech and door knock noise degraded by colored Gaussian noise (SNR=5dB) obtained by (2.17)-(2.18). 40

2.4 A realization of the test signal (speech signal plus keyboard typing plus babble noise) together with hand-marked and proposed VAD. 44

2.5 Probability of detection versus probability of false alarm (**Training**: SNR=10 dB; Stationary Noise: Babble; Transient Noise: Metronome; **Testing**: SNR=10 dB; Stationary Noise: Babble; Transient Noise: Metronome). 51

2.6 Probability of detection versus probability of false alarm (**Training**: SNR= 5 dB, Stationary Noise: Babble; Transient Noise: Keyboard Typing; **Testing**: SNR= 5 dB; Stationary Noise: Babble; Transient Noise: Keyboard Typing). . . 52

2.7 Probability of detection versus probability of false alarm (**Training**: SNR= 5 dB; Stationary Noise: Babble; Transient Noise: Keyboard Typing; **Testing**: SNR= 5 dB; Stationary Noise: White Gaussian; Transient Noise: Metronome) . 53

2.8 Probability of detection versus probability of false alarm (**Training**: SNR= 10 dB; Stationary Noise: Colored Gaussian; Transient Noise: Door Knock; **Testing**: SNR= 10 dB; Stationary Noise: Babble; Transient Noise: Metronome) 54

3.1 Comparison of the Algorithm 3.2 (second row) and the method proposed in [93] (third row): (a) One dimensional manifold embedded in R^3, (b)-(c) The embedding using the first two nondegenerate eigenvectors together and the estimated vector field, respectively. (d)-(e) The embedding using the first two nondegenerate eigenvectors together and the estimated vector field, respectively. 77

3.2 Comparison of the Algorithm 3.1 (a) and the method proposed in [27] (b) for embedding and vector field extension 78

3.3 Comparison of the Algorithm 3.1 (a) and the geometric harmonics (b) for function extension. 78

3.4 Affinity matrix (a) and the comparison of the proposed method and the the methods due to Joncas-Meila (presented in [93]) and Chen-Yang-Tang (presented in [22] for web page classification. 79

3.5 Affinity matrix (a) and the comparison of the proposed method and the the method due to Joncas-Meila (presented in [93]) for supervised system identification. 79

4.1 Normalized root mean square error (NRMSE) in function extension for different methods as a function of the parameter k. 96

4.2 Normalized root mean square error (NRMSE) in function extension for different methods as a function of the parameter k. 97

4.3 Normalized root mean square error (NRMSE) in function extension for different methods as a function of the parameter k. 97

4.4 Normalized root mean square error (NRMSE) in estimating the parameters of the AR part for different methods as a function of the parameter k. 98

4.5 Normalized root mean square error (NRMSE) in estimating the parameters of the MA part for different methods as a function of the parameter k. 98

4.6 Estimated reflection coefficient β_{x_1} versus the true value for different methods. . 99

4.7 Estimated reflection coefficient β_{x_2} versus the true value for different methods. . 99

4.8 Normalized root mean square error (NRMSE) in estimating the reflection coefficient β_{x_1} for different methods as a function of the parameter k. 100

4.9 Normalized root mean square error (NRMSE) in estimating the reflection coefficient β_{x_2} for different methods as a function of the parameter k. 100

5.1 The results of the RML method in estimating time varying parameters for three different SNRs. 114

6.1 (a) Spectrum of a typical clean speech signal together with the spectrum of a synthetic AR-GARCH process. (b) QQ-plot of the quantiles of speech signal versus the quantiles of a synthetic AR-GARCH process. 120

6.2 (a) Spectrum of a typical clean a speech signal together with the spectrum of a synthetic AR process. (b) QQ-plot of the quantiles of a speech signal versus the quantiles of a synthetic AR process with Gaussian residual. 121

6.3 (a) Clean speech signal together with hand-marked VAD. (b) Comparison of ROC curves of the proposed method with those of reference methods for Gaussian noise and SNR=5dB. 134

6.4 (a) Comparison of ROC curves of the proposed method with those of reference methods for factory noise and SNR=5dB. (b) Comparison of ROC curves of the proposed method with those of reference methods for babble noise and SNR=5dB. 135

6.5 (a) ROC curves of the proposed VAD method for different AR orders. (b) Comparison of ROC curves of the proposed method with those of reference methods in presence of a 60 Hz Additive Sinusoidal Component and SNR=5dB. 137

7.1 The conditional variance of the centered pixel is a weighted mean of the squared value of the neighboring pixels where the weights are symmetric (i.e. pixels dotted with the same color have the same weight). 143

7.2 The sample cross-correlation of: (a) η_L and y, (b) η_U and y, (c) η and y, and (d) η_L and η_U. 147

7.3 PDF of (a): $\hat{\boldsymbol{a}}_1$, (b): $\hat{\boldsymbol{a}}_2$, (c): $\hat{\boldsymbol{a}}_3$, (d): $\hat{\boldsymbol{a}}_4$ and (e): $\hat{\boldsymbol{a}}_5$ for different value of N for an $N \times N$ noncausal ARCH$(1,1)$ process. (f) Estimated normalized root mean square error in parameter estimation of an $N \times N$ noncausal ARCH$(1,1)$ process as a function of N. 156

7.4 ROC curves: comparison of detection performance between noncausal ARCH and GARCH modeling for synthetic data. 157

7.5 Sample side scan sonar image with a sea mine (sea mines are characterized by a bright line (the highlight or echo), corresponding to the scattering effect of the mine to the acoustic insonification, and a shadow behind it, corresponding to the blocking of sonar waves by the mine). These images each 16×16 pixel, are used as training image for obtaining the anomaly subspace. 159

7.6 A sample side-scan sonar image with artificial mine (a), true detection (b), generalized likelihood ratio L_{GLR} obtained by noncausal ARCH modeling (c) and GARCH modeling (d. 160

7.7 ROC curves: comparison of detection performance between noncausal ARCH and GARCH modeling for real data. 161

List of Tables

2.1 Proposed Voice Activity Detection Algorithm Based On Spectral Clustering Method 43

2.2 Elapsed Time in Second for Different Methods. 47

2.3 Voice Activity Detection Performance Comparison (**Training**:SNR=10 dB; Stationary Noise: Babble; Transient Noise: Metronome; **Testing**: SNR=10 dB; Stationary Noise: Babble noise; Transient Noise: Metronome). 48

2.4 Voice Activity Detection Performance Comparison (**Training**: SNR= 5 dB; Stationary Noise: Babble; Transient Noise: Keyboard Typing; **Testing**: SNR= 5 dB; Stationary Noise: Babble; Transient Noise: Keyboard Typing). 48

2.5 Voice Activity Detection Performance Comparison (**Training**: SNR= 5 dB; Stationary Noise: Babble; Transient Noise: Keyboard Typing; **Testing**: SNR= 5 dB; Stationary Noise: White Gaussian; Transient Noise: Metronome) 49

2.6 Voice Activity Detection Performance Comparison (**Training**: SNR= 10 dB; Stationary Noise: Colored Gaussian; Transient Noise: Door Knock; **Testing**: SNR= 10 dB; Stationary Noise: Babble; Transient Noise: Metronome) 49

2.7 Voice Activity Detection Performance Comparison (**Training**: SNR= 20 dB; Stationary Noise: Babble; Transient Noise: Door Knock; **Testing**: SNR= 5 dB; Stationary Noise: Colored Gaussian; Transient Noise: Keyboard typing) 50

3.1 Proposed Algorithm for Embedding and Function Extension on a Directed Graph 80

3.2 Proposed Algorithm for Finding the Embedding, the Sampling Density and the Vector Field Representing the Directed Graph 81

4.1 Proposed Algorithm for Extension of Band-Limited Functions on Homogenous Manifolds Using Diffusion Maps 90

5.1 Algorithm For ML Parameter Estimation of Noisy Complex GARCH Model Without Considering Stationarity Conditions 108
5.2 Algorithm for ML Parameter Estimation of Noisy Complex GARCH Model, Under Stationarity Conditions 109
5.3 MSE in Parameter Estimation for Different Methods (SNR_{in}=-10dB) 112
5.4 MSE in Parameter Estimation for Different Methods (SNR_{in}=-5dB) 112
5.5 MSE in Parameter Estimation for Different Methods (SNR_{in}=0.0dB) 113
5.6 MSE in Parameter Estimation for Different Methods (Colored Noise) (SNR_{in}=-4.44dB) 113
5.7 MSE in Parameter Estimation for RML Methods with Three Different SNR . . 114

6.1 Recursive ML Algorithm for Estimating the Parameters of a Noisy AR-GARCH Process 126
6.2 The Proposed VAD Algorithm Using AR-GARCH Modeling 131
6.3 MSE in Parameter Estimation and MSE in Estimation of $\sigma^2_{t|t-1}$ for Different SNR Levels 132
6.4 Parameter Values Used for the Proposed VAD 133
6.5 Probability of Detection Versus Probability of False Alarm for Different SNR Levels and Different Methods (White Noise) 135
6.6 Probability of Detection Versus Probability of False Alarm for Different SNR Levels and Different Methods (Factory Noise) 136
6.7 Probability of Detection Versus Probability of False Alarm for Different SNR Levels and Different Methods (Babble Noise) 136
6.8 Probability of Detection Versus Probability of False Alarm for Different SNR Levels and Different Methods (60 Hz Additive Sinusoidal Component) 137

7.1 Anomaly Detection Algorithm Using Noncausal AR-ARCH model 155

8.1 Proposed Algorithm for Estimating the Parameters of a Linearly Constrained Model Using Particle Methods 172
8.2 NRMSE in Parameter Estimation For Different Methods 174

Abstract

The field of speech signal processing has attracted significant research effort for several decades, but many aspects of it remain open and require further research. This dissertation addresses the theory and applications of geometric and model-based methods in signal processing.

This dissertation is divided into two parts. In the first part, we concentrate on geometric methods in signal processing. First, we address the voice activity detection problem. Speech/non-speech classification is an unsolved problem in speech processing and affects diverse applications including robust speech recognition, discontinuous transmission, real-time speech transmission on the Internet or combined noise reduction and echo cancellation schemes in the context of telephony. We address the problem of voice activity detection in presence of transient noise using geometric methods. More specifically, We use spectral clustering to perform this task. Embedding and function extension on directed graphs is the next topic presented in this thesis. We provide a graph embedding algorithm which is motivated by Laplacian type operator on manifold. We also introduce a Nyström type eigenfunctions extension which is used both for extending the embedding to new data points and to extend an empirical function on new data set. We also concentrate on the problem of extending a band-limited function defined on homogeneous manifold (i.e. the domain of the function is a homogeneous manifold embedded in the Euclidean space). The proposed method has a closed form solution and consists of matrix multiplication and inversion. As the size of data approaches infinity, the proposed method converges to the optimal solution as long as the function values are known on an appropriate sampling set.

In the second part of this thesis, we concentrate on the model-base signal processing. We choose the autoregressive conditional heteroscedasticity (ARCH) model and its generalization, the GARCH model. This model represents a powerful tool for the analysis and forecasting of volatility in financial markets. ARCH model is a statistical model which explicitly parametrizes

a time-varying conditional variance using squared absolute values, while considering volatility clustering and excess kurtosis (i.e. heavy-tailed distribution). We use this model along with the conventional autoregressive (AR) model and present a voice activity detection procedure based on the AR-GARCH modelling of speech signal in time domain.

The problem of anomaly detection in SONAR images is also addressed in this thesis. We introduce a novel model (i.e. Non-causal AR-ARCH model) for modelling the background of a SONAR image in wavelet domain. We also provide a deep analytical analysis of the proposed model. An efficient method for estimating the model parameters together with analytical analysis of this estimator are also provided.

The last topic addressed in our research is parameter estimation of the ARCH and GARCH models in presence of additive noise. We provide two different methods based on maximum likelihood estimation for performing this task. In the first method, we find the exact likelihood function of the noisy observations and find the maximum of this function using gradient-based methods. Here, we assume that both the process noise and the corrupting noise are distributed normally. Since these conditions might be violated in practical situations, we propose a method based on particle filters for maximum likelihood estimation of the parameters of the ARCH and the GARCH models. This method is based on gradient descend method and active set method for maximizing the likelihood function over parameters under stationarity constraints. The gradient of the likelihood function of observation given the parameters of the model, which is needed for gradient based optimization algorithm, is estimated using particle methods.

Notation

x	scalar
$\boldsymbol{x}$	column vector
$\boldsymbol{X}$	matrix
$x(n)$	time-domain signal
$(\cdot)^T$	transpose operation
$(\cdot)^H$	Hermitian
$\lvert x \rvert$	absolute value
$\lVert \boldsymbol{x} \rVert$	Euclidian norm
$\lvert \boldsymbol{X} \rvert$	determinant
$\boldsymbol{X}^{-1}$	matrix inverse
$\mathrm{E}\cdot$	expectation
∇	gradient
$\nabla \cdot \boldsymbol{r}$	divergence of the vector field $\boldsymbol{r}$
$\triangle$	Laplace-Beltrami operator
R^N	N-demential real-valued vector
C^N	N-demential complex-valued vector
∂	partial derivative
$\boldsymbol{a} \cdot \boldsymbol{b}$	inter product
$\xrightarrow{p}$	convergence in probability

Abbreviations

AR	Autoregressive
ARCH	Autoregressive Conditional Heteroscedasticity
AED	Acoustic Event Detection
AIC	Akaike Information Criterion
BEC	Back End Clipping
BIC	Bayesian Information Criterion
CFAR	Constant False Alarm Rate
CTSLS	Constrained Two Stage Least Squares
DWT	Discrete Wavelet Transform
EKF	Extended Kalman Filters
EM	Expectation-Maximization
FEC	Front End Clipping
GAECH	Generalized Autoregressive Conditional Heteroscedasticity
GMM	Gaussian Mixture Model
GMRF	Gauss Markov Random Field
HMM	Hidden Markov Model
IMCRA	Improved Minima Controlled Recursive Averaging
KLT	Karhunen-Loeve Transform
LLE	Local Linear Embedding
LMM	Linear Mixing Model
LRT	Likelihood Ratio Test
LSA	Log Spectral Amplitude
MFCC	Mel Frequency Cepstrum Coefficients
MFD	Matched Filter Detector
ML	Maximum Likelihood

MMSE	Minimum Mean Square Error
MSC	Mid Speech Clipping
MSD	Matched Subspace Detector
MSE Mean Square Error	
NDS	Noise Detected As Speech
NRMSE	Normalized Root Mean Square Error
PCA	Principle Component Analysis
pd	Probability Of Detection
pdf	Probability Density Function
pfa	Probability Of False Alarm
QML	Quasi-Maximum Likelihood
RML	Recursive Maximum Likelihood
ROC	Receiver Operating Curve
SA	Stochastic Approximation
SHT	Single Hypothesis Test
SNR	Signal To Noise Ratios
SONAR	Sound Navigation And Ranging
STFT	Short Time Fourier Transform
TSLS	Two Stage Least Squares
UKF	Unscented Kalman Filter
VAD	Voice Activity Detection
WLLN	Weak Low Of Large Numbers

Chapter 1

Introduction

This dissertation addresses the theory and applications of geometric and model-based methods in speech processing. Speech processing is of great interest in many hands-free communication systems. This field has attracted significant research effort for several decades, but many aspects remain open and require further research. In this Chapter, we describe the topics explored in this research, and indicate the scientific gaps that motivated our work. This research was directed at developing and analysing data driven and model-based algorithms for speech processing. Speech signals can be processed with two different approaches. In the first approach, which is data driven, the data is processed based on the geometry of raw data. Geometry of data can supply invaluable information regarding the distribution of the data in high dimensional space which consequently can be used in designing efficient algorithms for data analysis. Alternatively speech signal can be processed with another approach which is based on a statistical model. In this approach, speech signals are modelled with a statistical model and further processed based on the assumed model.

This thesis is divided into two distinct parts. The first part including Chapters 2 through 4, uses geometric methods for analysing data. More specifically, in Chapter 2, we develop a novel VAD algorithm based on spectral clustering methods. In Chapter 3, we propose a novel technique for finding the graph embedding and function extension for directed graphs. In Chapter 4, we address the problem of extending a band-limited function defined on homogeneous manifold (i.e. the domain of the function is a homogeneous manifold embedded in the Euclidean space).

The second part of the thesis including Chapters 5 through 8, starts with introducing a

parameter estimation method for estimating the parameters of complex GARCH model from noisy data. Then in Chapter 6, we introduce a voice activity detector based on AR-GARCH modelling of the speech signal in time domain. In Chapter 7, we introduce a new statistical model, i.e. two dimensional (2D) non-causal AR-ARCH model, for modelling the background clutter in SONAR images in the wavelet domain and utilize this model for anomaly detection in SONAR images. Finally, in Chapter 8, we propose a new method based on particle filters for ML estimation of the parameters of ARCH and GARCH models in presence of additive noise.

1.1 Geometric Tools for Data Analysis

In this section, we have a brief review on geometric tools for data analysis and model-based speech signal processing and discuss the basic theory and methods which were used during this research. We also briefly outline the topics investigated in this dissertation.

In recent years, there has been a growing effort to develop data analysis algorithms based on the geometry of the acquired raw data. An important branch of this approach, termed kernel methods or manifold learning, aims at discovering the underlying structure in data-sets as a precursor to other types of processing. These include ISOMAP [123], locally linear embedding (LLE) [103], Laplacian eigenmap [14], and diffusion maps [28]. These data analysis approaches have recently become very popular in machine learning and data mining applications, e.g. image segmentation [46], classification of tissue types [34], lipreading from video [71], organizing and clustering texts [72], etc. The ability in capturing the underlying structure of data, which usually has a much lower dimension than the data itself, has made these methods appropriate tools for data dimensionality reduction. These algorithms exhibit two major advantages over classical data dimensionality reduction methods (such as principal component analysis (PCA)): they are non-linear, and they preserve local structures. The first aspect is essential, as most of the time the data points in their original form, do not lie on a linear manifold. The second point is the fact that in many applications, distances of points that are far apart from each other are meaningless. Therefore, these distances do not need to be preserved. In what follows, we briefly review the diffusion maps since we intend to use them as the framework in our research. For a detailed discussion on diffusion maps, see [32].

1.1.1 A Brief Introduction to Diffusion Maps

Let $X \in \mathrm{C}^d$ be a high dimensional finite data-set where C is the set of complex numbers and d is the dimension of X. In addition, suppose that we are given a kernel $k : X \times X \to \mathrm{R}$ which is symmetric and nonnegative (i.e. $k(x_i, x_j) = k(x_i, x_j) \geq 0 \; \forall x_i, x_j \in X$). This kernel represents some notion of similarity between points of X as it describes the relationship between pairs of points in this set. A popular way of representing the data is to build a weighted graph $G = (V, E)$ where V is the set of the vertices and E is the set of the edges of the graph. Each vertex v_i in this graph represents a data point x_i and each edge e_{ij} between two vertices v_i and v_j, carries a non-negative weight equal to $k(x_i, x_j)$.

Let $\boldsymbol{P}$ be the transition matrix of a Markov chain on X, constructed from kernel k as follows:

$$\boldsymbol{P}_{i,j} = \frac{k\left(x_i, x_j\right)}{d\left(x_i\right)} \tag{1.1}$$

$$d\left(x_i\right) = \sum_{j=1}^{N} k\left(x_i, x_j\right) \tag{1.2}$$

where $\boldsymbol{P}_{i,j}$ is the ij-th element of $\boldsymbol{P}$ and N is the number of data-points in set X. From a data analysis point of view, the reason for studying this Markov chain is that the matrix $\boldsymbol{P}$ contains geometric information about the data-set X. Indeed, the transitions that it defines, directly reflect the local geometry defined by the immediate neighbors of each node in the graph of the data. $\boldsymbol{P}_{i,j}$ represents the probability of transition in one time step from x_i to x_j and is proportional to the $k(x_i, x_j)$. For $t \geq 0$, the probability of transition from x_i to x_j in t time steps is given by $\boldsymbol{P}^t_{i,j}$ (the ij-th element of the t-th power $\boldsymbol{P}$).

A classical way to describe the powers of an operator is to employ the language of spectral theory, namely eigenvectors and eigenvalues. Let $\{\lambda_l; l = 1, 2, ..., N\}$ and $\{\psi_l; l = 1, 2, ..., N\}$ be the set of the eigenvalues and eigenvectors of $\boldsymbol{P}$ respectively, such that $1 = \lambda_1, \geq |\lambda_2| \geq |\lambda_3| \geq \cdots | \geq \lambda_N|$. The diffusion distance between x_i to x_j is defined as follows:

$$D^t_{i,j} = \sum_{l=1}^{N} \left(\boldsymbol{P}^t_{i,k} - \boldsymbol{P}^t_{j,k}\right)^2 \tag{1.3}$$

It can be shown that the diffusion distance can be computed using the eigenvalues and eigenvectors of $\boldsymbol{P}$ as follows:

$$D^t_{i,j} = \sum_{k=1}^{N} \lambda_l^{2t} \left(\psi^i_l - \psi^j_l\right)^2 \tag{1.4}$$

where ψ_l^i is the i-th element of ψ_l. Since the eigenvalues $\lambda_2, \lambda_3, \cdots$, tend to zero and have a modulus strictly less than one, the above sum can be computed to a preset accuracy $\delta > 0$ with a smaller number of terms. So if

$$L = \max\{l \in \mathrm{N} s.t. |\lambda_l|^t \geq |\lambda_2|^t\}$$

then

$$D_{i,j}^t = \sum_{k=1}^{L} \lambda_l^{2t} \left(\psi_l^i - \psi_l^j\right)^2 \tag{1.5}$$

up to relative precision δ. Hence, a family of diffusion maps can be defined as follows:

$$\Psi_t(x_i) \triangleq \begin{bmatrix} \lambda_2^{2t}\psi_2(xi) \\ \lambda_3^{2t}\psi_3(xi) \\ \vdots \\ \lambda_L^{2t}\psi_L(xi) \end{bmatrix}. \tag{1.6}$$

Each component of $\Psi_t(x)$ is termed diffusion coordinate. The map $\Psi_t : X \to \mathrm{R}^L$ embeds the data-set into a Euclidean space of L dimensions in which the Euclidean between two points in this space is equal to the diffusion distance in original space up to relative precision δ. This fact implies that diffusion maps can be used as a tool for dimensionality reduction.

1.1.2 Voice Activity Detection in Presence of Transient Noise Using Spectral Clustering

Speech/non-speech classification is an unsolved problem in speech processing and affects diverse applications including robust speech recognition [65, 97], discontinuous transmission [16], real-time speech transmission on the Internet [100] or combined noise reduction and echo cancelation schemes in the context of telephony [13]. Elementary methods for voice activity detection such as G.729 standard [16], calculate line spectral frequencies, full-band energy, low-band energy (<1 kHz), and zero-crossing rate. Each frame is then simply classified using a fixed decision boundary in the space defined by these features. Smoothing and adaptive correction can be applied to improve the estimate. Although these methods have acceptable performance when applied to clean signals, their performance essentially degrades in noisy environments even in moderately high signal to noise ratios (SNRs). To overcome this shortcoming, several

statistical model-based VAD algorithms have been proposed in the last two decades. Sohn et al. [111] 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 likelihood ratio test (LRT). Following their work, many researchers tried to improve the performance of model-based VAD algorithms by assuming different statistical models for speech signals, see [21, 37, 61, 84, 96, 108] and references therein. While these methods have superior performances in presence of stationary noise over the elementary methods, their performances degrade significantly in presence of transient noise such as coughing, sneezing, keyboard typing, and door knocking sounds. This means that with high probability, these sounds are detected as speech.

VAD is usually a preprocessing step in speech processing applications such as speech or speaker recognition. A straightforward application of VAD would be an automatic camera steering task. Suppose a scenario in which there exist multiple speakers with a camera assigned to each of them (a popular example can be videoconferencing). The camera must be steered to the dominant speaker automatically. While stationary noise can be treated very well using a statistical mode-based method, transient noise could be very annoying [127]. This means that a silent speaker might be identified as a dominant speaker while he/she is just typing or there is a knock on the door. Hence, finding a VAD algorithm which is robust to transient noise would be of practical interest.

VAD can be regarded as an acoustic event detection (AED) task which detects some acoustical event including transient noise, e.g., door knocking, footsteps, etc. Current most prominent works in AED reflect the aim of bringing the most successful technologies of speech recognition to the field. Zhou et al. [137] implemented a hidden Markov model (HMM)-based AED system with lattice rescoring using a feature set selected by AdaBoost based approach. Haung et al. [60] improved AED via audio-visual intermediate integration using generalizable visual features. Using optical flow based spatial pyramid histograms, they proposed a method for representing the highly variant visual cues of the acoustic events. Espi et al. [45] introduced the usage of spectro-temporal fluctuation features in a tandem connectionist approach, modified to generate posterior features separately for each fluctuation scale and then combine the streams to be fed to a classic Gaussian mixture model–hidden Markov model (GMM-HMM) procedure. Voice activity detection can also be regarded as a clustering problem, in which the

goal is to classify the input signal into speech absence and speech presence frames. Hence, after choosing an appropriate feature space, one can use a clustering algorithm to obtain a VAD algorithm. Among different clustering methods, spectral clustering has recently become one of the most popular modern clustering algorithms. It is simple to implement, can be solved efficiently by standard linear algebra software, and very often outperforms the traditional clustering algorithms such as the k-means algorithm. Recently, spectral clustering has been utilized by several authors in signal processing applications such as image segmentation [107, 133], speech separation [12], and clustering of biological sequence data [92] just to name a few.

In Chapter 2, we present a novel voice activity detection algorithm using spectral clustering. In particular, we use a normalized spectral clustering algorithm [77] to cluster the Mel-frequency cepstrum coefficients (MFCC) of the received signal into two different clusters, i.e., speech presence and speech absence. The clustering problem can be done using GMM. However, fitting a GMM to high dimensional data generally requires a great amount of training data, and as the number of Gaussian mixtures is increased, we need more and more training data to fit the GMM to high dimensional data. The fact that the distribution of natural data, like speech and transient noise is non-uniform and concentrates around low-dimensional structures [63], motivates us to exploit the shape (geometry) of the distribution for efficient learning. Among different dimensionality reduction techniques, kernel eigenmap methods such as local linear embedding [103], Laplacian eigenmaps [14], Hessian eigenmaps [41], and diffusion maps [32] (just to name a few) have recently attracted much research attention. These algorithms exhibit two major advantages over classical dimensionality reduction methods (such as principal component analysis or classical multidimensional scaling): They are nonlinear, and they preserve local structures. The first aspect is essential as most of the time, in their original form, the data points do not lie on linear manifolds. The second point is related to the fact that in many applications, distances of points that are far apart are meaningless, and therefore need not to be preserved. The main idea of these methods is to use the dominant eigenvectors of Laplacian of the similarity matrix as the new lower dimension representation of the data.

Our proposed algorithm is a supervised learning algorithm. One must train the system before it can be used. Training data is used for estimating the parameters of the kernel used in computation of the similarity matrix. The data is also used in finding two Gaussian mixture models for modeling the first two eigenvectors of the Laplacian of the similarity matrix cor-

responding to the first two leading eigenvalues of normalized Laplacian matrix. This means that we model the low dimensional representation of the original data (i.e. MFCC) using two different GMMs, one for each cluster. Upon receiving new unlabeled data, the optimum parameters of the kernel are utilized to find the similarity between the new data and the training set in order to find the low dimensional representation of new data. Using the GMMs obtained in the training step, the likelihood ratio is computed, and the final VAD is obtained by comparing that likelihood ratio to a threshold.

1.1.3 Embedding and Function Extension on Directed Graph

Recent advances in geometric based methods of data mining and machine learning lead to efficient algorithms for lots of applications such as dimensionality reduction, function extension, classification and clustering, just to name a few. Most of these methods are graph based techniques. Graphs offer an advantageous compromise between their simplicity, interpretability and their ability to express complex relationships between data points. The core idea in such algorithms is to construct a weighted graph on data points such that each vertex of the graph represents a data point, and a weighted edge, connecting two vertices to each other, represents the similarity between the two corresponding data points. In the context of networks (e.g., social or networks), the data naturally lead themselves to graph modeling [59]. The graph based representation of data combined with Markov chain techniques exhibits extremely successful results. The main idea here is based on the fact that the eigenvectors of Markov matrices can be regarded as coordinates on the data set. Among vast techniques incorporating Markov chain methods in data processing, kernel eigenmap methods have attracted much research attention, recently. The algorithmic consequence of these methods are local linear embedding (LLE) [99], Laplacian eigenmaps [14], Hessian eigenmaps [40], local tangent space alignment [134] and diffusion maps [32].

An important application of geometric based method is out of sample function extension, in which an empirical function is extended to unlabeled data. Examples of this application include clustering, regression, and classification. In these applications, usually a large amount of data is involved and the only way to perform these tasks is to subsample the data set $\overline{X}$ in order to reduce the size of the problem, process the new set X, and then extend the result to the original

data $\overline{X}$. Coifman and Lafon [27], proposed a geometric harmonics procedure, inspired from the Nyström method, to perform this task. More specifically, they assumed that the similarity between the data points is given by a symmetric positive semi-definite kernel. Then it is shown that the eigenfunctions of the integration operator defined by this kernel, form an orthogonal basis for the space of squared integrable functions defined on $\overline{X}$ (i.e. $L^2(\overline{X})$). In order to extend a function defined on the set X to the data set $\overline{X}$, first the eigenfunctions computed on X are extended to the data set $\overline{X}$ using the Nyström method. The function is then approximated as the weighted sum of these extended eigenfunctions. Kushnir et al. [70] and Singer et al. [110] introduced a method for parameterizing high dimensional data into its independent physical parameters, which enables the identication of the parameters and a supervised extension of the re-parameterization to new observations. In their work, a novel diffusion processes was used, utilizing only the small observed set, that approximates the isotropic diffusion on the parametric manifold. They utilized Nyström-type extension of the embedding of that small observed data-set to the embedding into the independent components on a much larger data-set.

In almost all of these kernel eigenmaps based methods, the similarity between points is given by a symmetric positive semi-definite kernel. In some practical applications the similarity between points is not necessarily symmetric. Among several examples in which the similarity between points has not to be necessarily symmetric, one can mention document classification based on citation graphs [53], web information retrieval based on hyperlink structure, and protein clustering based on pair-wise alignment scores [92]. Chen et al. [22], proposed an algorithm for embedding vertices on directed graphs to vector spaces. This algorithm explores the inherent pair-wise relation between vertices of the directed graph by using transition probability and the stationary distribution of Markov random walks, and embeds the vertices into vector spaces preserving such relation optimally. Recently, Perrault-Joncas and Meilă [93] proposed an algorithm based on analysis of Laplacian type operators and their continuous limit as generators of diffusions on a manifold. They modeled the observed graph as a sample from a manifold endowed with a vector field, and designed an algorithm that separates and recovers the features of this process: the geometry of the manifold, the data density, and the vector field. The most important shortcoming of these methods is not providing a straightforward procedure to extend the embedding to new points in case only the distances of the new point to the original data points are known, which encountered in applications. In [31], Coifman and Hirn intro-

duced a simple procedure for construction of a bi-stochastic kernel for an arbitrary data set that is derived from an asymmetric affinity function. The affinity function measures the similarity between points in test set and some reference set.

These geometrically based algorithms have been applied in various signal processing applications. In a pioneering work, Shi and Malik [107] used spectral methods for image segmentation. Later many researchers have used geometrically based methods in applications such as image clustering [46, 133], image completion [52], speech enhancement in presence of transient noise [120], voice activity detection in presence of transient noise [85], linear and nonlinear independent component analysis [109, 110], parametrization of linear systems [121], and single channel source localization [119].

In Chapter 3, we propose a novel technique for embedding a directed graph to Euclidian space. We model the observed data as samples from a manifold where the similarity between the points is given by an asymmetric kernel. This asymmetric kernel is modeled utilizing a vector field, and we design an algorithm that separates and recovers the geometry of the manifold, the data density, and the vector field. We further provide a simple Nyström extension procedure which allows us to extend both the embedding and the estimated vector field to new data points.

1.1.4 Out-of-Sample Extension of Band-Limited Functions on Homogenous Manifolds Using Diffusion Maps

Classical Nyquist [106] theorem states that, a bandlimited function defined on the reals can be determined from its value on equally spaced points of reals, if the sampling rate is high enough. The value of the function on points other than sampling points, can be exactly calculated using SINC interpolator. Schoenberg [102] used cardinal splines for the reconstruction formula. There it is shown that a bandlimited function can be reconstructed from its values sampled at high enough rate (Nyquist rate) as accurately as needed using cardinal splines of sufficiently high degree. This result was further generalized to the case of nonuniform sampling by Lyubarskii and Madych [78]. More specifically, they showed that a bandlimited function $f(x)$, whose Fourier transform is compactly supported between $[-\pi,\pi]$ can be completely reconstructed using spline functions, from its samples $f(x_n)$ taken at sampling point x_n, in the case when the functions $\exp(jx_n\omega)$, form a Riesz basis for $L_2([-\pi,\pi])$. Pesenson [94] generalized

the concept of bandlimited functions to the case that the domain of the function is a homogenous manifold and introduced the spectral entire functions of exponential type and Lagrangian splines on homogeneous manifolds. He also showed that on manifolds, the reconstruction of irregularly sampled spectral entire functions of exponential type (from now on bandlimited functions) by splines is possible, as long as the distance between points from a sampling sequence is small enough. Recently, using a totally different point of view, Coifman and Lafon [33] proposed a simple scheme, based on the Nyström method for extending empirical functions defined on a set X to a larger set $\bar{X}$. The extension process involves the construction of a specific family of functions termed geometric harmonics. These functions constitute a generalization of the prolate spheroidal wave functions of Slepian in the sense that they are optimally concentrated on X. Although being a powerful tool for function extension, this scheme does not make use of unlabeled data to learn the manifold.

The central dogma for studying the problem of function extension on manifolds is that the distribution of natural data is non-uniform and concentrates around low-dimensional structures. The shape (geometry) of the distribution can be exploited for efficient learning. As a justification for manifold assumption of natural data, see [63] in speech processing and [41] in computer vision. Geometrically based methods have been used in applications such as image clustering [46, 133], image completion [52], speech enhancement in presence of transient noise [120], voice activity detection in presence of transient noise [85], linear and nonlinear independent component analysis [109, 110], parametrization of linear systems [121], and single channel source localization [119]. Function extension has a lot of applications in machine learning. Among them one can name clustering, classification and dimensionality reduction. In some applications, the manifold and the function to be extended on it, are both unknown and just some labeled data (the points that the value of function is known on them, i.e. sampling points) and unlabeled data (the points that the value of function on them must be determined, i.e. interpolation points) are available. This means that the sampling theorem on manifolds [94] cannot be used directly to extend the function to unlabeled data because in this method, the manifold is assumed to be known. In this case the geometric harmonics can be easily applied for function extension but as mentioned before, the invaluable unlabeled data are not used.

In order to be able to use a sampling theorem on manifolds, one needs to completely know the manifold. This means that the Laplace-Beltrami operator on the manifold must be known

and could be computed for every function. Many manifold learning algorithms have been introduced during the last decade, among them one can name isomap [123], Locally-linear embedding (LLE) [103], Laplacian eigenmaps [14] and diffusion maps [32]. Diffusion maps leverage the relationship between heat diffusion and a random walk on a graph. The heat diffusion on manifold, is the diffusion process whose infinitesimal generator is the Laplace-Beltrami operator. In [32], an analogy is drawn between the diffusion operator on a manifold and a Markov transition matrix operating on functions defined on the graph whose nodes were sampled from the manifold. It is also shown that one can approximate the Laplace-Beltrami operator using appropriately normalized Markov transition matrix.

In Chapter 4, we propose a novel technique for extending a bandlimited function defined on a manifold. More specifically, we use diffusion maps as a tool for manifold learning and approximating the Laplace-Beltrami operator on a manifold. Next we use sampling theorem of bandlimited functions on manifolds [94] to extend the function onto the interpolation points.

1.2 Model-based speech signal processing

Statistical model-based speech processing has a history of more than two decades. This approach has been utilized by many researchers in a variety of speech processing applications: speech enhancement, voice activity detection and speaker localization, just to name a few. In [43, 44], Ephraim and Malah used Gaussian distribution for modeling the speech signal in the short-time Fourier transform (STFT) domain. They derived a minimum mean squares error (MMSE) estimator of the spectral amplitude and log-spectral amplitude (LSA) of the speech signal and proposed a speech enhancement algorithm. About twenty years later, Cohen and Berdugo used the LSA estimator along with the speech presence probability in each frequency bin and proposed the well-known optimally modified log-spectral amplitude (OM-LSA) algorithm for speech enhancement [26]. In [24], speech enhancement using supper-Gaussian distribution and noncausal a priori signal to noise ratio (SNR) estimation is addressed. More specifically, the speech signals in the STFT domain given the spectral variance are assumed to have Laplacian or Gamma distribution, and the MMSE estimator of the spectral amplitude of the speech signal was obtained under each of these assumptions.

Autoregressive conditional heteroscedasticity (ARCH) model was first introduced by Engle

in 1982 [42]. This model represents a powerful tool for the analysis and forecasting of volatility in financial markets. ARCH model is a statistical model which explicitly parameterizes a time-varying conditional variance using squared absolute values, while considering volatility clustering and excess kurtosis (i.e. heavy-tailed distribution). This feature of ARCH models enables them to be applied to numerous economic and financial data to model unpredictability; the strong dependence of the instantaneous variability of a time series on its own past. Another feature of the ARCH process is that it is a white process. So, in some applications it may be useful for modeling heavy-tailed white processes.

Since 1982, the ARCH model has been expanded by several author: GARCH introduced by Bollerslev [19], Complex GARCH introduced by Cohen [25] and Markov switching GARCH model introduced by Abramson and Cohen [5] just to name a few. Each of these generalizations has its roots in the seminal work of Engle [42]. In each of the above mentioned works authors introduced a new model based on the ARCH model, such that their model best fit the data to be processed.

Recently, Cohen used the generalized autoregressive conditional heteroscedasticity (GARCH) models to model the speech signals in the STFT domain and proposed an algorithm for speech enhancement [25]. Abramson and Cohen utilized the Markov-switching GARCH (MS-GARCH) model for speech enhancement in the STFT domain [3]. They also used the MS-GARCH model in a variety of speech processing applications such as dual-microphone speech dereverberation [7] and single microphone audio source separation [6].

The model-based approach has been also utilized in voice activity detection (VAD). VAD is an integral part of many speech communication systems. Examples of such systems are audio conferencing, hands-free telephony [49] and discontinuous speech transmission [16]. During the last few decades, many researchers have dealt with this issue. In statistical model-based VADs, the likelihood ratio test (LRT) is applied to a set of hypotheses [21, 96, 111]. In all of these methods, the speech signal is transformed to the STFT domain and different statistical models are applied to the speech data in the STFT domain. For instance, Sohn et al. [111] and Ramirez et al. [96] assumed that the spectral coefficients of the noise and the speech signals are complex Gaussian random variables. Chang et al. [21] utilized the complex Laplacian and Gamma probability density functions (pdfs) to model the distribution of the speech spectrum. Shin et al. [108] applied the generalized gamma distribution to model the distribution of the

clean speech spectrum.

Although the statistical model-based speech processing algorithms have shown acceptable performance in speech processing, their performance is highly dependent on the choice of the model. This means that finding a more appropriate model for speech signals helps in improving the performance of the underlying speech processing system. In this research, we intend to utilize a different model for modeling the speech signal in the time domain and present a novel technique for voice activity detection.

1.2.1 Simultaneous Parameter Estimation and State Smoothing of Complex GARCH Process in the Presence of Additive Noise

Parameter estimation is the backbone of all model-based signal processing algorithms. Having variability clustering and heavy tail property, autoregressive conditional heteroscedasticity (ARCH) and generalized ARCH (GARCH) models have recently been used in speech signal modeling. In speech enhancement in the short time Fourier transform (STFT) domain, the most important quantity which is to be estimated correctly is the spectral variance, since having the true value of this quantity, one can use the minimum mean square error (MMSE) [43] or log spectral amplitude (LSA) estimators, which are optimum in MMSE and LSA sense, respectively, in order to estimate the speech spectral component. Cohen [25] modeled the speech signal in the STFT domain as a complex GARCH process and used this model for speech enhancement. He showed that the time varying variance of the speech signal can be estimated using a GARCH model with Gaussian innovations. Furthermore, decision-directed method of Ephraim and Malah [43] can be derived using GARCH modeling. In his work, the GARCH model is used to estimate the spectral variance of the speech signals. Tahmasbi et al. [118] used the GARCH model for the voice activity detection (VAD). Abdolahi et al. [2] used the parameters of the GARCH model for speech recognition in Persian isolated digits. In all of the above-mentioned applications, it is assumed that the parameters of the GARCH model are known or can be estimated from a clean database.

Parameter estimation methods for ARCH and GARCH models, such as quasi-maximum likelihood (QML) [129], two stage least squares (TSLS) [20], and constrained two stage least squares (CTSLS) [86], rely on the clean data. Since these models are mostly used in econo-

metric where data can be assumed noise free, there was no effort to estimate the parameters of these models in presence of additive noise. In many speech signal processing algorithms, these methods are inefficient since the additive noise is inevitable. Hence, new methods for estimating ARCH and GARCH parameters in presence of noise are required. Poyiadjis et al. [95] introduced a method for ML parameter estimation of general state-space models using particle methods. The drawback of their method is its very high computational load, which makes this approach inapplicable to speech signal processing.

In Chapter 5, we propose a new ML based estimation method for estimating the parameters of GARCH model in presence of additive noise. We find the probability density function (pdf) of the observation data given the parameters and compute the maximum of this function. Specifically, we compute the gradient vector and the Hessian matrix of the probability density function of the data, and subsequently find the maximum of the probability density function numerically using the Newton method.

1.2.2 AR-GARCH in Presence of Noise: Parameter Estimation and its Application to Voice Activity Detection

Voice activity detection (VAD) is an integral part of all speech communication systems. Examples of such systems are audio conferencing, hands-free telephony [49] and discontinuous speech transmission [16]. During the last few decades, many researchers have dealt with this issue. The modern VADs are derived from statistical model-based algorithms in which the likelihood ratio test (LRT) is applied to a set of hypotheses. In all of these methods, the speech signal is transformed to the short-time Fourier transform (STFT) domain and different models are applied to the speech data in the STFT domain. For instance, Sohn et al. [111] and Ramirez et al. [96] assumed that the spectral coefficients of the noise and speech signal are complex Gaussian random variables. Chang et al. [21] utilized the complex Laplacian and Gamma probability density functions (pdfs) to model the distributions of the speech and noise spectra. Shin et al. [108] applied the generalized gamma distribution to model the distribution of the clean speech spectrum. Davis et al. [37] proposed the statistical VAD method that makes no assumption about the distributions of the speech spectra.

Recently, Cohen [25] modeled the speech signal in the STFT domain as a complex GARCH

process and used this model for speech enhancement. Solvang et al. [112] used the AR-GARCH model for VAD. In their work, they represent the AR part of the AR-GARCH model with a state-space to obtain the appropriate linear prediction error series. By applying the GARCH model to the residuals, they estimate the conditional variance sequences corresponding to the voice activity parts. To detect voice activity, they establish an appropriate threshold for the conditional variance sequences. However, their method is not computationally efficient and cannot be used in real time speech signal processing. Furthermore, they assume that the model is identified from frames with speech being present, but in the application at hand these frames are unknown to the user in advance.

In Chapter 6, we present a new VAD using AR-GARCH modeling of the speech signal in the time domain. This model relies on the fact that speech signals in the time domain can be modeled as AR processes [15] and also demonstrate both the variability clustering and the heavy-tail behavior, that are the main properties of GARCH processes. Our VAD is based on the LRT, so we need to compute the likelihood function of the observations under the speech presence and absence hypotheses. The time varying variance of the speech signal needed for computing the likelihood function under speech presence hypothesis is estimated using the AR-GARCH model. To be able to use this model, one should know the parameters of the model or estimate them from the available data. Since the available data is often corrupted by additive noise, we present a novel technique for parameter estimation of the AR-GARCH model in presence of additive noise. To compute the likelihood function of the observations under speech presence and absence hypotheses, we need to know the variance of the additive noise, which might be time varying. The improved minima controlled recursive averaging (IMCRA) [23] method has been modified and utilized for estimating the variance of the corrupting noise in the time domain.

1.2.3 Two Dimensional Noncausal AR-ARCH Model: Stationary Conditions, Parameter Estimation and its Application to Anomaly Detection

Image anomaly detection is the process of extracting a small number of clustered pixels, which are different from the background. The type of image and the characteristics of anomalies are

application dependent. Among the applications one can name detection of targets in images, detection of defects in silicon wafers, detection of mine features in side-scan sonar and detection of tumors in medical imaging. Target detection in radar and sonar imagery is a challenging problem due to the large variability in background clutter and in object appearance. The detection of sea-mines, for example, involves addressing the varying shape of the ocean surface and its vegetation [39]. In most cases, lethal targets must be detected with nearly 100% reliability. False detections may not be disastrous but might slow down the demining process.

Anomaly detection algorithms generally consist of some or all of the following stages: selection of an appropriate feature space, selection of a statistical model for the selected feature space and selection of a detection algorithm.

A proper selection of a feature space, which allows distinction of anomalies from the clutter, is an important part of an anomaly detection algorithm. Feature space selection methods can be classified into two major groups: Image pixel feature space and transform feature space. In *image pixel feature space*, the feature space is created based on the image pixels themselves. Kazantsev et al. [67] introduced a feature space based on two circular concentric windows W_1 and W_2 with radii R_1 and R_2, respectively, $R_1 < R_2$. A similar approach is taken by Schweizer and Moura [104]. In their approach two concentric rectangles serve as the moving window. For further discussion on this kind of feature space selection see [67] and [104]. In *transform feature space*, the feature space is created based on the transformed image. There exist two common transforms which are mainly used for anomaly detection: Karhunen-Loeve transform (KLT) and discrete wavelet transform (DWT). The KLT is used to transform an n dimensional vector space into an m dimensional vector space, where $m < n$, such that the mean-square magnitude of the error resulting from representing the n dimensional vector using only m dimensions is minimized. The KLT is also used to remove correlation between features. DWT is recently used in variety of applications in image signal processing. Among them, the most important one is image compression and coding [128]. The theory of two dimensional DWT is well developed and can be found in classic textbooks, (see [128]).

Once the feature space is chosen, one must find an appropriate statistical model to describe the natural clutter in the selected feature space domain. Popular models for modeling the natural clutter are Gaussian or extensions of Gaussian, selected because of their mathematical tractability. Ashton [11] performed subpixel anomaly detection in multispectral infrared imagery using

Gaussian distribution for clutter. Stein et al. [114] used a Gaussian mixture model (GMM) for modeling hyperspectral imagery. Other extensions such as linear mixing model (LMM) and Gauss Markov random field (GMRF) are used by several authors for modeling the clutter. See [130] for a good review of multi-resolution Markov models for signal and image processing. Recently, Noiboar and Cohen [91] used *causal* GARCH model for anomaly detection in sonar images. The causality assumption incorporated into the GARCH model in [91], is unnatural for images. Developing a non-causal statistical model characterized by a heavy tailed distribution and innovation clustering leads to an improvement in clutter modeling by reducing the dependency on image orientation. As a result, such a model, may reduce the false alarm rate for a given detection rate.

By anomaly detection we mean classifying a region of an image as anomaly or background with the assumption of low-probability anomalies. There are several anomaly detector among them one can name single hypothesis test (SHT) [48], matched filter detector (MFD) and its adaptive version [79], matched subspace detector (MSD) [101] and its adaptive version [68].

In Chapter 7, we present a new statistical model to describe the natural clutter in DWT domain. This model generalizes the method presented in [91] by using a noncausal ARCH instead of GARCH model.

1.2.4 ARCH and GARCH Parameter Estimation in Presence of Additive Noise Using Particle Methods

As discussed before, parameter estimation is an important step in model-based signal processing. Additive noise which is inevitable in practical application results in excessive error in model parameter estimation if it is not taken into account. The ML method we present in Chapter 5, assumes that both the process noise and the corrupting noise are both distributed normally. This might not be true in practical application. In Chapter 8, we propose a gradient based technique for estimating the parameters of ARCH and GARCH models in presence of additive noise. More specifically, we use a gradient descent method together with active set method to obtain the maximum of the likelihood function. The gradient of the likelihood function of noisy observations given the parameters which is needed for gradient based optimization algorithm, is obtained by particle method.

1.3 Thesis Structure

This thesis is organized ad follows.

In Chapter 2, we develop a novel VAD algorithm based on spectral clustering methods. We propose a VAD technique which is a supervised learning algorithm. This algorithm divides the input signal into two separate clusters (i.e. speech presence and speech absence frames). We use labeled data in order to adjust the parameters of the kernel used in spectral clustering methods for computing the similarity matrix. The parameters obtained in the training stage together with the eigenvectors of the normalized Laplacian of the similarity matrix and Gaussian mixture model (GMM) are utilized to compute the likelihood ratio needed for voice activity detection. Simulation results demonstrate the advantage of the proposed method compared to conventional statistical model-based VAD algorithms in presence of transient noise.

In Chapter 3, we propose a novel technique for finding the graph embedding and function extension for directed graphs. We assume that the data points are sampled from a manifold and the similarity between the points is given by an asymmetric kernel. We provide a graph embedding algorithm which is motivated by Laplacian type operator on manifold. We also introduce a Nystöm type eigenfunctions extension which is used both for extending the embedding to new data points and to extend an empirical function on new data set. For extending the eigenfunctions to new points, we assume that only the distances of the new points to the labeled data are given. Simulation results demonstrate the performance of the proposed method recovering the geometry of data and extending a function on new data points.

In Chapter 4, we address the problem of extending a bandlimited function defined on homogenous manifold (i.e. the domain of the function is a homogenous manifold embedded in the Euclidian space). We solve this problem in the general case where the manifold is unknown. We assume that we have sufficient labeled data to learn the function. We also assume that we have enough data to learn the manifold (i.e. to learn the domain of the function). The proposed method has a closed form solution and consists of matrix multiplication and inversion. As the size of data approaches infinity, the proposed method converges to the optimal solution as long as the function values are known on an appropriate sampling set. Simulation results demonstrate the advantage of the proposed method over geometric harmonics, widely used as a function extension method.

In Chapter 5, we present a new ML estimation procedure for parameter estimation of complex GARCH$(1,1)$ model in presence of the additive noise, and a new procedure for estimating the hidden states of the model in the MMSE sense. The method computes the log-likelihood function together with its gradient vector and Hessian matrix, and uses the Newton method (or gradient projection method under stationarity conditions) to numerically find the maximum of the log-likelihood function. We also present an RML method for estimating the time varying parameters. Simulation results show improvement with respect to not considering the noise effect in the estimation of the parameters. We consider two situations. The first one is the general case where the SNR is unknown. In this situation, the proposed method estimates the variance of the noise together with the parameters and the hidden states of the model. The second situation, which is a special case of the first one, assumes that the variance of the additive noise is known and just estimates the parameters of the model. As expected, the second case yields better performance over the first one because of the additional information employed. The issue how to handle colored noise is also discussed in this Chapter.

In Chapter 6, we present a new VAD based on AR(p)-GARCH $(1,1)$ modeling of the speech signal. We also introduced a novel procedure based on the ML estimation method for parameter estimation of the AR$(p)-$GARCH $(1,1)$ model in presence of additive noise. We present an adaptive version of parameter estimation method, namely, the RML method. In this method, upon receiving a new sample, we update the likelihood function together with its gradient vector and used the steepest descent method to numerically find the maximum of the log-likelihood function. The proposed VAD is obtained by comparing the likelihood ratio to a threshold. The likelihood ratio of the observations was computed using the estimated variance of the speech signal, which was obtained by the RML algorithm. We use the IMCRA method for estimating the variance of the additive noise, which can be used under nonstationary conditions. This enabled our VAD to follow variations in the variance of the additive noise. Simulation results demonstrate the high performance of the proposed method and particularly its advantage in nonstationary noise environments.

In Chapter 7, we introduce a new statistical model, noncausal AR-ARCH model. We give sufficient stationarity condition and introduce a novel technique based on TSLS method to estimate the parameters of the model. We also present an anomaly detection algorithm based on AR-ARCH modeling of the background. The detection algorithm is matched subspace de-

tector and the clutter model is noncausal AR-ARCH model. This model is an extension of the model used in [91] for anomaly detection. The advantage of the proposed model over the causal GARCH model is its low computational load and its invariability to the orientation of the image. Simulation results demonstrate the improved performance of the proposed method.

In Chapter 8, we propose a new method based on particle filters for maximum likelihood (ML) estimation of the parameters of autoregressive conditional heteroscedasticity (ARCH) and generalized autoregressive conditional heteroscedasticity (GARCH) models. Our method is based on gradient descend method and active set method for maximizing the likelihood function over parameters under stationarity constraints. The gradient of the likelihood function of observation given the parameters of the model, which is needed for gradient based optimization algorithm, is estimated using particle methods. Simulation results show the advantage of the proposed method over competing techniques.

Finally in Chapter 9, we summarize the work and provide directions for future research.

1.4 List of Publications

The Chapters of this thesis are based on the following publications:

Chapter 2 is based on:

S. Mousazadeh and I. Cohen "*Voice Activity Detection in Presence of Transient Noise Using Spectral Clustering*" IEEE Trans. Audio, Speech and Language Processing, Vol. 21, Number 6, June 2013, pp. 1261-1271.

Chapter 3 is based on:

S. Mousazadeh and I. Cohen "Embedding and Function Extension on Directed Graph" Submitted to Signal Processing.

Chapter 4 is based on:

S. Mousazadeh and I. Cohen "Simultaneous Parameter Estimation and State Smoothing of Complex GARCH Process in the Presence of Additive Noise" Signal Processing, Vol. 90, Issue 11, Nov. 2010, pp. 2947-2953.

Chapter 5 is based on:

S. Mousazadeh and I. Cohen "*Simultaneous Parameter Estimation and State Smoothing of Complex GARCH Process in the Presence of Additive Noise*" Signal Processing, Vol. 90, Issue 11, Nov. 2010, pp. 2947-2953.

Chapter 6 is based on:

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, Issue 4, May 2011, pp. 916-926.

Chapter 7 is based on:

S. Mousazadeh and I. Cohen "*Anomaly Detection in Sonar Images Based on Wavelet Domain Noncausal AR-ARCH Random Field Modeling*" Proc. 26th IEEE Convention of Electrical and Electronics Engineers in Israel, IEEEI-2010, Eilat, Israel, 17-20 November 2010.

S. Mousazadeh and I. Cohen "Two dimensional noncausal AR-ARCH model: stationary conditions, parameter estimation and its application to anomaly detection" Signal Processing, Vol. 98, May 2014, pp 322336.

Chapter 8 is based on:

S. Mousazadeh and I. Cohen "*ARCH and GARCH Parameter Estimation in Presence of Additive Noise Using Particle Methods*" Proc. 38th IEEE Internat. Conf. Acoust. Speech Signal Process., ICASSP-2013, Vancouver, Canada, May 26-31, 2013, pp. 6279-6282.

Chapter 2

Voice Activity Detection in Presence of Transient Noise Using Spectral Clustering

2.1 Introduction

Speech/non-speech classification is an unsolved problem in speech processing and affects diverse applications including robust speech recognition [65, 97], discontinuous transmission [16], real-time speech transmission on the Internet [100] or combined noise reduction and echo cancelation schemes in the context of telephony [13]. Elementary methods for voice activity detection such as G.729 standard [16], calculate line spectral frequencies, full-band energy, low-band energy ($<$1 kHz), and zero-crossing rate. Each frame is then simply classified using a fixed decision boundary in the space defined by these features. Smoothing and adaptive correction can be applied to improve the estimate. Although these methods have acceptable performance when applied to clean signals, their performance essentially degrades in noisy environments even in moderately high signal to noise ratios (SNRs). To overcome this shortcoming, several statistical model-based VAD algorithms have been proposed in the last two decades. Sohn et al. [111] assumed that the spectral coefficients of the noise and speech signal can be modelled as complex Gaussian random variables, and developed a VAD algorithm based on the likelihood ratio test (LRT). Following their work, many researchers tried to improve the performance of model-based VAD algorithms by assuming different statistical models for speech signals, see [21, 37, 61, 84, 96, 108] and references therein. While these methods have superior performances in presence of stationary noise over the elementary methods, their performances

degrade significantly in presence of transient noise such as coughing, sneezing, keyboard typing, and door knocking sounds. This means that with high probability, these sounds are detected as speech.

VAD is usually a preprocessing step in speech processing applications such as speech or speaker recognition. A straightforward application of VAD would be an automatic camera steering task. Suppose a scenario in which there exist multiple speakers with a camera assigned to each of them (a popular example can be videoconferencing). The camera must be steered to the dominant speaker automatically. While stationary noise can be treated very well using a statistical mode-based method, transient noise could be very annoying [127]. This means that a silent speaker might be identified as a dominant speaker while he/she is just typing or there is a knock on the door. Hence, finding a VAD algorithm which is robust to transient noise would be of practical interest.

VAD can be regarded as an acoustic event detection (AED) task which detects some acoustical event including transient noise, e.g., door knocking, footsteps, etc. Current most prominent works in AED reflect the aim of bringing the most successful technologies of speech recognition to the field. Zhou et al. [137] implemented a hidden Markov model (HMM)-based AED system with lattice rescoring using a feature set selected by AdaBoost based approach. Haung et al. [60] improved AED via audio-visual intermediate integration using generalizable visual features. Using optical flow based spatial pyramid histograms, they proposed a method for representing the highly variant visual cues of the acoustic events. Espi et al. [45] introduced the usage of spectro-temporal fluctuation features in a tandem connectionist approach, modified to generate posterior features separately for each fluctuation scale and then combine the streams to be fed to a classic Gaussian mixture model–hidden Markov model (GMM-HMM) procedure. Voice activity detection can also be regarded as a clustering problem, in which the goal is to classify the input signal into speech absence and speech presence frames. Hence, after choosing an appropriate feature space, one can use a clustering algorithm to obtain a VAD algorithm. Among different clustering methods, spectral clustering has recently become one of the most popular modern clustering algorithms. It is simple to implement, can be solved efficiently by standard linear algebra software, and very often outperforms the traditional clustering algorithms such as the k-means algorithm. Recently, spectral clustering has been utilized by several authors in signal processing applications such as image segmentation [107, 133],

speech separation [12], and clustering of biological sequence data [92] just to name a few.

In this chapter, we present a novel voice activity detection algorithm using spectral clustering. In particular, we use a normalized spectral clustering algorithm [77] to cluster the Mel-frequency cepstrum coefficients (MFCC) of the received signal into two different clusters, i.e., speech presence and speech absence. The clustering problem can be done using GMM. However, fitting a GMM to high dimensional data generally requires a great amount of training data, and as the number of Gaussian mixtures is increased, we need more and more training data to fit the GMM to high dimensional data. The fact that the distribution of natural data, like speech and transient noise is non-uniform and concentrates around low-dimensional structures [63], motivates us to exploit the shape (geometry) of the distribution for efficient learning. Among different dimensionality reduction techniques, kernel eigenmap methods such as local linear embedding [103], Laplacian eigenmaps [14], Hessian eigenmaps [41], and diffusion maps [32] (just to name a few) have recently attracted much research attention. These algorithms exhibit two major advantages over classical dimensionality reduction methods (such as principal component analysis or classical multidimensional scaling): They are nonlinear, and they preserve local structures. The first aspect is essential as most of the time, in their original form, the data points do not lie on linear manifolds. The second point is related to the fact that in many applications, distances of points that are far apart are meaningless, and therefore need not to be preserved. The main idea of these methods is to use the dominant eigenvectors of Laplacian of the similarity matrix as the new lower dimension representation of the data.

Our proposed algorithm is a supervised learning algorithm. One must train the system before it can be used. Training data is used for estimating the parameters of the kernel used in computation of the similarity matrix. The 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 normalized Laplacian matrix. This means that we model the low dimensional representation of the original data (i.e. MFCC) using two different GMMs, one for each cluster. Upon receiving new unlabeled data, the optimum parameters of the kernel are utilized to find the similarity between the new data and the training set in order to find the low dimensional representation of new data. Using the GMMs obtained in the training step, the likelihood ratio is computed, and the final VAD is obtained by comparing that likelihood ratio to a threshold.

The rest of this chapter is organized as follows. In Section 2.2, we formulate our problem and introduce a novel VAD for online processing. Simulation results and performance comparison are presented in Section 2.3. Finally, we conclude the chapter in Section 2.4.

2.2 Problem Formulation

In this section, we propose our voice activity detection method, which is based on spectral clustering. The basic idea behind spectral clustering method is to use several eigenvectors of the normalized Laplacian of the similarity matrix as a new low dimension representation of the high dimension data points. Clustering is generally performed on this new representation of the data points using a conventional (weighted) k-means algorithm [12]. Here we introduce a novel technique for clustering the data based on GMM modelling of the eigenvectors of the normalized Laplacian of the similarity matrix.

Every clustering problem consists of the following three main stages: selecting an appropriate feature space, choosing a metric as a notion of similarity between data-points, and selecting the clustering algorithm. In what follows, we discuss each of these stages for voice activity detection in more detail.

2.2.1 Feature Selection

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

$$y(n) = x_{\text{sp}}(n) + x_{\text{tr}}(n) + x_{\text{st}}(n). \tag{2.1}$$

The goal is to determine whether there exists speech signal in a given time frame (each approximately 16-20 msec long). Here we choose absolute value of MFCCs and the arithmetic mean of the log-likelihood ratios for the individual frequency bins as our feature space. More specifically, let $\boldsymbol{Y}_m(t,k)$ $(t = 1,2,...,N; k = 1,2,...,K_m)$ and $\boldsymbol{Y}_s(t,k)$ $(t = 1,2,...,N; k = 1,2,...,K_s)$ be the absolute value of the MFCC and the STFT coefficients in a given time frame, respectively. MFCC and the STFT coefficients are computed in K_m and K_s frequency bins, respectively.

Then, each frame is represented by a (K_m+1)-dimension column vector defined as follows:

$$\boldsymbol{Y}(:,t) = \begin{bmatrix} \boldsymbol{Y}_m(:,t) \\ \Lambda_t \end{bmatrix} \tag{2.2}$$

where $\boldsymbol{Y}_m(:,t)$ is the t-th column of $\boldsymbol{Y}_m$, and Λ_t is the arithmetic mean of the log-likelihood ratios for the individual frequency bands in frame t, which is given by:

$$\Lambda_t = \frac{1}{K_s}\sum_{k=1}^{K_s}\left(\frac{\gamma_k(t)\xi_k(t)}{1+\xi_k(t)} - \log\left(1+\xi_k(t)\right)\right) \tag{2.3}$$

where $\xi_k(t) = \lambda_s(t,k)/\lambda_N(t,k)$ is called *a priori* SNR, which can be estimated using decision-directed method [43], and $\lambda_s(t,k)$ is the variance of speech signal in the k-th frequency bin of the t-th frame; $\gamma_k(t) = |\boldsymbol{Y}_s(t,k)|^2/\lambda_N(t,k)$ is called the *a posteriori* SNR, ε is the kernel width obtained during the training phase, and $\lambda_N(t,k)$ is the variance of stationary noise in t-th time frame and k-th frequency bin, which can be estimated from training data (if there exist sequences consisting of only stationary noise) or by improved minima controlled recursive averaging (IMCRA) [23]. The reason behind choosing this feature space for each frame is as follows. The likelihood ratio has been long exploited as a feature for voice activity detection in presence of stationary noise [21, 37, 84, 108, 111]. The Mel-frequency cepstrum coefficient is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear Mel scale of frequency. MFCCs are commonly used as features in speech recognition systems. Combining these two features appropriately would be a suitable feature space for voice activity detection in presence of transient noise. See the next section for further discussion.

2.2.2 Clustering Algorithm

A popular way for representing the data is to build a similarity graph, which is a weighted graph $G = (V,E)$, where V is the set of the vertices and E is the set of the edges of the graph. Each vertex v_i in this graph represents a data point $\boldsymbol{Y}(:,i)$. Each edge e_{ij} between two vertices v_i and v_j carries a non-negative weight $\boldsymbol{W}(i,j) \geq 0$, which is a measure of similarity between the corresponding points. We assume that the graph G is an undirected one (i.e. $\boldsymbol{W}(i,j) = \boldsymbol{W}(j,i)$). A similarity matrix $\boldsymbol{W}$ is a matrix whose (i,j)-th element equals to $\boldsymbol{W}(i,j)$.

Using the concept of subspace comparison, Bach and Jordan [12] proposed a spectral clustering algorithm using the eigenvectors of matrix $\boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$ where $\boldsymbol{D}$ is a diagonal matrix

whose i-th diagonal element equals to $\sum_{j=1}^{N} \boldsymbol{W}(i,j)$ (i.e. $\boldsymbol{D} = \mathrm{diag}\left(\boldsymbol{W}\mathbf{1}\right)$ where $\mathbf{1}$ is a column vector of ones). More specifically, let K be the number of clusters and $\boldsymbol{U}$ be a matrix consisting of the first K eigenvectors of $\boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$ corresponding to K's largest eigenvalues of $\boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$. The clustering is done either by running a weighted k-means algorithm on $\boldsymbol{U}$ where each point is represented by a row of $\boldsymbol{U}$ or by running a k-means algorithm on $\boldsymbol{V} = \boldsymbol{D}^{1/2}\boldsymbol{U}\left(\boldsymbol{U}^T\boldsymbol{D}\boldsymbol{U}\right)^{-1}$ where each point is represented by a row of $\boldsymbol{V}$. The most important drawback of this method is that it does not prepare a toll for controlling the trade-off between probability of false alarm and probability of detection. We will deal with this issue by using GMM modelling of the eigenvectors (see testing algorithm below for further discussion).

The most important part of a spectral clustering algorithm is the calculation of the similarity matrix. Although the definition of the similarity between points is an application and data dependent, a popular way of defining the similarity matrix is to use a Gaussian kernel as follows:

$$\boldsymbol{W}(i,j) = \exp\left(-\frac{\|\boldsymbol{Y}(:,i) - \boldsymbol{Y}(:,j)\|_2^2}{\sigma}\right) \tag{2.4}$$

where $\boldsymbol{Y}(:,i)$ is the i-th data point. The selection of σ is commonly done manually. Ng et al. [90] suggested selecting σ automatically by running their clustering algorithm repeatedly for a number of values of σ and selecting the one that provides the least distorted clusters. In [58] and [35], it was suggested to automatically set the scale by examining a logarithmic scale of the sum of the kernel weights without computing the spectral decomposition of the transition matrix. Zelnik-Manor et al. [133] suggested calculating a local scaling parameter σ_n for each data point instead of selecting a single scaling parameter σ. The above mentioned methods are somehow heuristic or hard to implement because of high computational load. Bach and Jordan [12] introduced a method for estimating the parameters of the kernel (not necessarily Gaussian kernel) based on minimization of a cost function that characterizes how close the eigenstructure of the similarity matrix $\boldsymbol{W}$ is to the true partition.

Our voice activity detection algorithm is a supervised learning one. As a consequence, one must utilize training data in order to adjust the parameters of the algorithm and use those parameters for clustering unlabelled data. In the next two subsections, we illustrate how each of these stages works.

Learning Algorithm

In this section, we introduce our learning algorithm based on the method presented in [12]. 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 as follows. Let $x^{\ell}_{\text{sp}}(n), x^{\ell}_{\text{tr}}(n), x^{\ell}_{\text{st}}(n)$ 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}). We build the ℓ-th training sequence, $\boldsymbol{Y}^{\ell} \in \mathrm{R}^{K_m+1\times 3N^{\ell}}$, as follows. Let

$$x^{\ell}_1(n) = x^{\ell}_{\text{sp}}(n) + x^{\ell}_{\text{st}}(n), \tag{2.5}$$

$$x^{\ell}_2(n) = x^{\ell}_{\text{tr}}(n) + x^{\ell}_{\text{st}}(n), \tag{2.6}$$

$$x^{\ell}_3(n) = x^{\ell}_{\text{sp}}(n) + x^{\ell}_{\text{tr}}(n) + x^{\ell}_{\text{st}}(n), \tag{2.7}$$

and let $\boldsymbol{Y}^{\ell}_1$, $\boldsymbol{Y}^{\ell}_2$, and $\boldsymbol{Y}^{\ell}_3$ be the feature matrix extracted using (6.27) and (2.3) from $x^{\ell}_1(n)$, $x^{\ell}_2(n)$, and $x^{\ell}_3(n)$, respectively. Then, the ℓ-th training data is obtained by concatenating these matrices as follows:

$$\boldsymbol{Y}^{l} = \left[\boldsymbol{Y}^{\ell}_1 \vdots \boldsymbol{Y}^{\ell}_2 \vdots \boldsymbol{Y}^{\ell}_3\right]. \tag{2.8}$$

A typical training sequence is depicted in Figure 2.1.

For each of these training sequences, we compute the indicator matrix of the partitions $\boldsymbol{C}^{\ell} \in \mathrm{R}^{3N^{\ell}\times 4}$ using

$$\boldsymbol{C}^{\ell}_{ij} = \begin{cases} \boldsymbol{C}^{\ell}_{i1} = 1 - \boldsymbol{C}^{\ell}_{i4} & \text{Only stationary noise} & 1 < i \leq N^{\ell} \\ \boldsymbol{C}^{\ell}_{i2} = 0 & \text{Transient plus stationary noise} & 1 < i \leq N^{\ell} \\ \boldsymbol{C}^{\ell}_{i3} = 0 & \text{Speech plus transient plus stationary noise} & 1 < i \leq N^{\ell} \\ \boldsymbol{C}^{\ell}_{i4} = \chi(P(\boldsymbol{X}^{\ell}_{\text{sp}}(:,i)) > T_{\text{sp}}) & \text{Speech plus stationary noise} & 1 < i \leq N^{\ell} \\ \boldsymbol{C}^{\ell}_{i1} = 1 - \boldsymbol{C}^{\ell}_{i2} & \text{Only stationary noise} & N^{\ell} < i \leq 2N^{\ell} \\ \boldsymbol{C}^{\ell}_{i2} = \chi(P(\boldsymbol{X}^{\ell}_{\text{tr}}(:,i)) > T_{\text{tr}}) & \text{Transient plus stationary noise} & N^{\ell} < i \leq 2N^{\ell} \\ \boldsymbol{C}^{\ell}_{i3} = 0 & \text{Speech plus transient plus stationary noise} & N^{\ell} < i \leq 2N^{\ell} \\ \boldsymbol{C}^{\ell}_{i4} = 0 & \text{Speech plust stationary noise} & N^{\ell} < i \leq 2N^{\ell} \\ \boldsymbol{C}^{\ell}_{i1} = 1 - (\boldsymbol{C}^{\ell}_{i2} \oplus \boldsymbol{C}^{\ell}_{i3} \oplus \boldsymbol{C}^{\ell}_{i4}) & \text{Only stationary noise} & 2N^{\ell} < i \leq 3N^{\ell} \\ \boldsymbol{C}^{\ell}_{i2} = \chi(P(\boldsymbol{X}^{\ell}_{\text{tr}}(:,i)) > T_{\text{tr}}) \otimes \chi(P(\boldsymbol{X}^{\ell}_{\text{sp}}(:,i)) < T_{\text{sp}}) & \text{Transient plus stationary noise} & 2N^{\ell} < i \leq 3N^{\ell} \\ \boldsymbol{C}^{\ell}_{i3} = \chi(P(\boldsymbol{X}^{\ell}_{\text{tr}}(:,i)) > T_{\text{tr}}) \otimes \chi(P(\boldsymbol{X}^{\ell}_{\text{sp}}(:,i)) > T_{\text{sp}}) & \text{Speech plus transient plus stationary noise} & 2N^{\ell} < i \leq 3N^{\ell} \\ \boldsymbol{C}^{\ell}_{i4} = \chi(P(\boldsymbol{X}^{\ell}_{\text{tr}}(:,i)) < T_{\text{tr}}) \otimes \chi(P(\boldsymbol{X}^{\ell}_{\text{sp}}(:,i)) < T_{\text{sp}}) & \text{Speech plus stationary noise} & 2N^{\ell} < i \leq 3N^{\ell} \end{cases} \tag{2.9}$$

where $\boldsymbol{C}^{\ell}_{ij}$ is the (i,j)-th element of $\boldsymbol{C}^{\ell}$, $\chi(.)$ is an indicator function that equals to one if its argument is true and zero otherwise, and T_{sp} and T_{tr} are speech and transient noise thresholds and are chosen as the maximum value of threshold such that thresholding the speech or transient

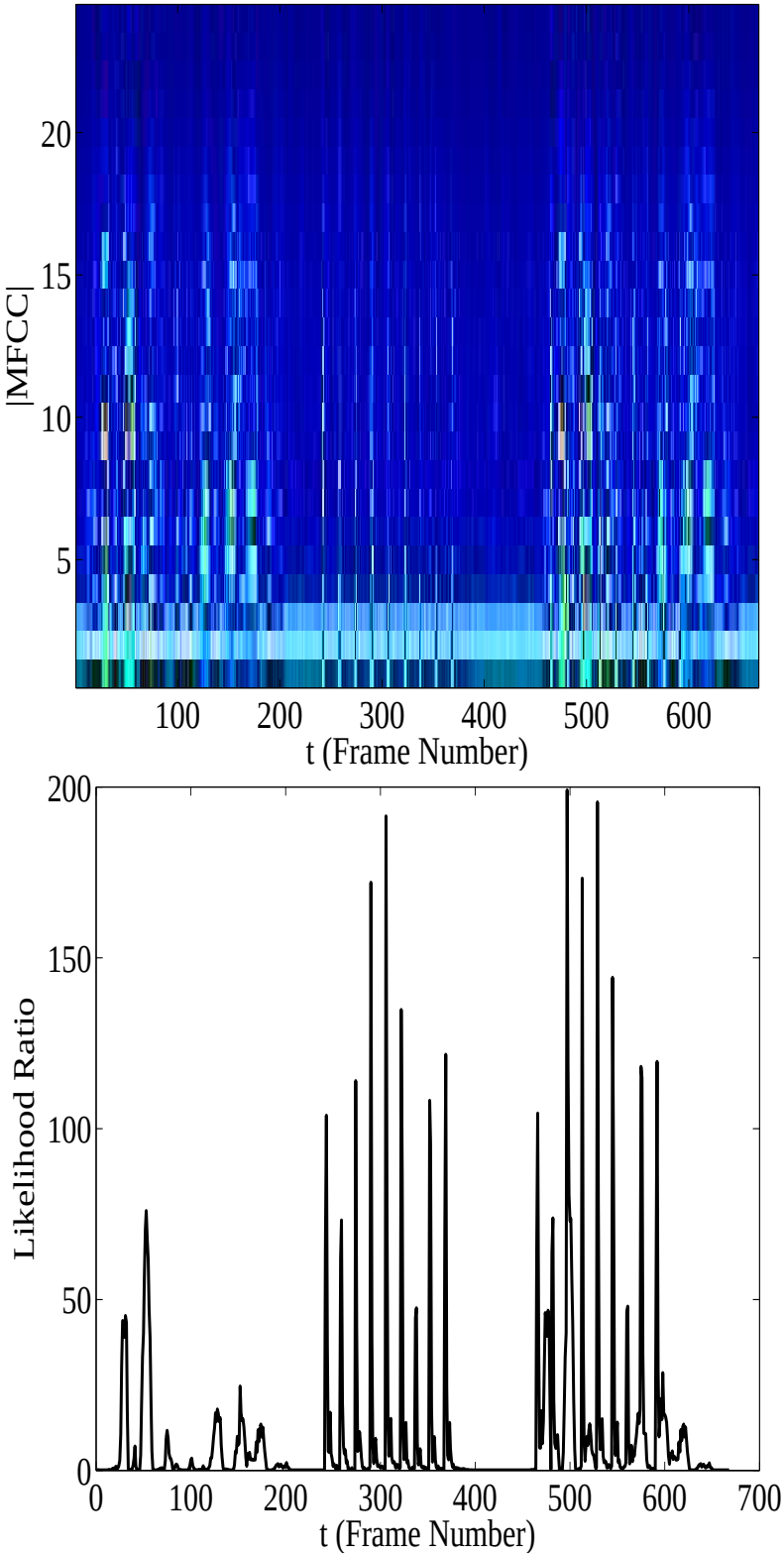


Figure 2.1: A typical training sequence consisting of speech signal and keyboard stoke noise (transient noise) corrupted by babble noise (absolute value of MFCC (right) likelihood ratio (left)).

noise has no significant auditory effect, $\oplus$ and $\otimes$ are logical **OR** and logical **AND** operators, respectively. $P(.)$ is a power calculation operator defined by:

$$P(\boldsymbol{X}_{\text{sp}}^{\ell}(:,i)) = \frac{1}{K_s}\sum_{k=1}^{K_s} \|\boldsymbol{X}_{\text{sp}}^{\ell}(k,i)\|_2^2 \tag{2.11}$$

$$P(\boldsymbol{X}_{\text{tr}}^{\ell}(:,i)) = \frac{1}{K_s}\sum_{k=1}^{K_s} \|\boldsymbol{X}_{\text{tr}}^{\ell}(k,i)\|_2^2. \tag{2.12}$$

where $\boldsymbol{X}_{\text{sp}}^{\ell}(:,i)$ and $\boldsymbol{X}_{\text{tr}}^{\ell}(:,i)$ are the STFT coefficients of $x_{\text{sp}}^{\ell}(n)$ and $x_{\text{tr}}^{\ell}(n)$ in the i-th frame, respectively.

The last and most important object to be defined, in order to use the training algorithm presented in [12], is the parametric similarity matrix, i.e., $\boldsymbol{W}_{\boldsymbol{\theta}}^{\ell}$. For our problem, we define this matrix as follows:

$$\boldsymbol{W}_{\boldsymbol{\theta}}^{\ell}(i,j) = \exp\left(\sum_{p=-P}^{P} -\alpha_p \boldsymbol{Q}(i+p,j+p)\right) \tag{2.13}$$

$$\boldsymbol{Q}(i,j) = \|\boldsymbol{Y}_m^{\ell}(:,i)\left(1-\exp(-\Lambda_i^{\ell}/\varepsilon)\right) - \boldsymbol{Y}_m^{\ell}(:,j)\left(1-\exp(-\Lambda_j^{\ell}/\varepsilon)\right)\|_2^2 \tag{2.14}$$

where $\boldsymbol{\theta} = [\varepsilon, \alpha_{-P}, \alpha_{-P+1}, \cdots, \alpha_{P-1}, \alpha_P] \in \mathrm{R}^{2P+2}$ is the vector of parameters, $\boldsymbol{Y}_m^{\ell}(:,i)$ and Λ_i^{ℓ} are the absolute value of the MFCC and the likelihood ratio of the ℓ-th training sequence in the i-th frame, respectively, and $\|.\|_2$ is the Euclidian norm of a vector. The reason behind choosing this weight function is discussed in the following paragraph.

For designing an appropriate weight matrix, we have taken the following two points into consideration. The first one was the similarity between two individual frames, and the second one was the effect of neighboring frames on deciding whether a specific frame contains speech or transient noise. Combining these two features (i.e. MFCC and likelihood ratio) as in (2.13)-(2.14), results in a good metric as a similarity notion between two frames for voice activity detection in presence of transient noise. More specifically, if there exists speech signal or transient noise in a specific frame, the value of likelihood ratio is large (see Figure 2.1 (right)); hence, the exponential term in (2.14) approximately equals to zero, and the feature for that frame will be approximately the MFCCs. On the other hand, if a specific frame consists of only stationary noise, then the likelihood ratio will be small, and the exponential term in (6.27) approximately equals to one. Consequently, the feature vector will approximately be equal to zero vector for those frames that only contain stationary noise. Considering the MFCCs of a typical training sequence as depicted in Figure 2.1 (left), it is apparent that there might exist

two frames in the speech part (left part of the figure) and the transient noise part (middle part of the figure) that are very similar to each other (in the sense that their Euclidean distance is small) but belong to two different clusters. The characteristic that distinguishes the frames containing speech from those frames containing transient noise is that the neighboring frames of a specific speech frame are almost the same, which is not true for transient noise. Choosing the weight function as in (2.13)-(2.14), guarantees small similarity between two frames from different classes (speech and transient noise) even if they are very similar to each other (in the Euclidean sense), because of the large distance between neighboring frames. Upon defining the parametric weight function, the parameters can be obtained by solving the following optimization problem [12]:

$$\boldsymbol{\theta}^{opt} = \arg\min_{\theta} \frac{1}{L} \sum_{\ell=1}^{L} F(\boldsymbol{W}_{\boldsymbol{\theta}}^{\ell}, \boldsymbol{C}^{\ell}) \tag{2.15}$$

$$F(\boldsymbol{W}, \boldsymbol{C}) = \frac{1}{2} \left\| \boldsymbol{\Upsilon}\boldsymbol{\Upsilon}^T - \boldsymbol{D}^{1/2}\boldsymbol{C}(\boldsymbol{C}^T\boldsymbol{D}\boldsymbol{C})^{-1}\boldsymbol{C}^T\boldsymbol{D}^{1/2} \right\|_F^2 \tag{2.16}$$

where L is the number of training sequence, $(\cdot)^T$ denotes transpose of a vector or a matrix, and $\boldsymbol{\Upsilon}$ 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* [54]. In practice, we use the gradient method (e.g. *fminunc* or *fmincon* functions in Matlab®) to solve this minimization problem.

Testing Algorithm

A testing algorithm aims to cluster the unlabeled data. The most straightforward way to perform clustering using spectral methods into K disjoint clusters is to use the parameters obtained by the learning algorithm, construct the similarity matrix $\boldsymbol{W}$, compute the K eigenvectors of $\boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$ corresponding to the first K largest eigenvalues (denoted by $\boldsymbol{U}$), and run weighted k-means algorithm on $\boldsymbol{U}$ or k-means algorithm on $\boldsymbol{V} = \boldsymbol{D}^{1/2}\boldsymbol{U}\left(\boldsymbol{U}^T\boldsymbol{D}\boldsymbol{U}\right)^{-1}$ [12]. This method has two major drawbacks. First, this method can only be used for batch processing (offline processing) of data. The second and more important one is that, this method does not allow the user to control the tradeoff between the probability of false alarm and the probability of detection. Every detection algorithm must be equipped with a tool such that one can increase the probability of detection (probably) by increasing the probability of false alarm. In order to

overcome these two shortcomings, we utilize the extension method proposed in [121] based on the fact that two test points are similar if they see the training data similarly, and the likelihood ratio test as our decision rule. In order to compute the likelihood ratio, we use GMM to model the eigenvectors of normalized Laplacian matrix. In what follows, we discuss these two issues in more detail.

Let $\boldsymbol{W}^{\ell}_{\boldsymbol{\theta}^{opt}}$ be the similarity matrix of ℓ-th training sequence and $\boldsymbol{U}^{\ell} \in \mathrm{R}^{3N^{\ell}\times 2}$ be a matrix consisting of the two eigenvectors of $\boldsymbol{D}^{\ell^{-1/2}}\boldsymbol{W}^{\ell}\boldsymbol{D}^{\ell^{-1/2}}$ corresponding to the first two largest eigenvalues. Let the column concatenation of $\boldsymbol{U}^1$ through $\boldsymbol{U}^L$ be

$$\boldsymbol{U} = \left[(\boldsymbol{E}^1 \odot \boldsymbol{U}^1)^T, \cdots, (\boldsymbol{E}^{\ell} \odot \boldsymbol{U}^{\ell})^T, \cdots, (\boldsymbol{E}_L \odot \boldsymbol{U}^L)^T\right]^T \tag{2.17}$$

$$\boldsymbol{E}^{\ell} = \sqrt{\boldsymbol{C}^{\ell}\,\mathrm{diag}\left(\mathbf{1}_{1\times N^{\ell}}\boldsymbol{C}^{\ell}\right)\mathbf{1}_{4\times 2}}; \quad \ell = 1,2,\cdots,L \tag{2.18}$$

where $\odot$ is term by term multiplication, $\mathrm{diag}(\boldsymbol{a})$, is a diagonal matrix whose diagonal is vector $\boldsymbol{a}$ and $\mathbf{1}_{m\times n}$ is an m by n matrix of ones. This normalization of the matrices $\boldsymbol{U}^1$ through $\boldsymbol{U}^L$ is due to a possible different number of points in the same cluster of different training sequences. Because of sign ambiguity in computation of eigenvectors, each of these eigenvectors is computed such that the mean of each cluster (noise only cluster or speech cluster) is as close as possible to the mean of each cluster of the first training sequence. More specifically, we compute the mean of low dimensional representation of each of the two clusters in the first training sequence and choose the sign of the eigenvectors corresponding to the remaining training sequence, such that their means are close to the means of the clusters in the first training sequence. We have selected this approach instead of combining all training sequences as a single training sequence because of computational load and memory usage. Combining all training sequence as a single sequence leads to a very large similarity matrix that cannot be handled computationally. This method is in some sense equivalent to ignoring the similarity between each training sequence, which is a widespread approach for sparsifying the similarity matrix [12]. Figures 2.2 and 2.3 show the scatter plot of the new representation of a typical learning sequence. These figures are obtained from five training sequences each approximately 10 seconds long. The similarity matrix is calculated using optimum parameters obtained by solving the optimization problem in (2.15). These figures are obtained in the case where the stationary noise was colored Gaussian noise, and the transient noise was a door knock. Figure 2.2 is the low dimensional representation of the training data obtained by concatenating the training sequences to a large sequence. Figure

2.3 is the low dimensional representation of the training data obtained by the aforementioned method (i.e. equations 2.17-2.18). These figures are experimental justification for the proposed approximation in computation of eigenvectors. It is also apparent that there exist four separated clusters (i.e. only stationary noise, transient plus stationary noise, speech plus transient plus stationary noise, and speech plus stationary noise), two of which contain speech signal.

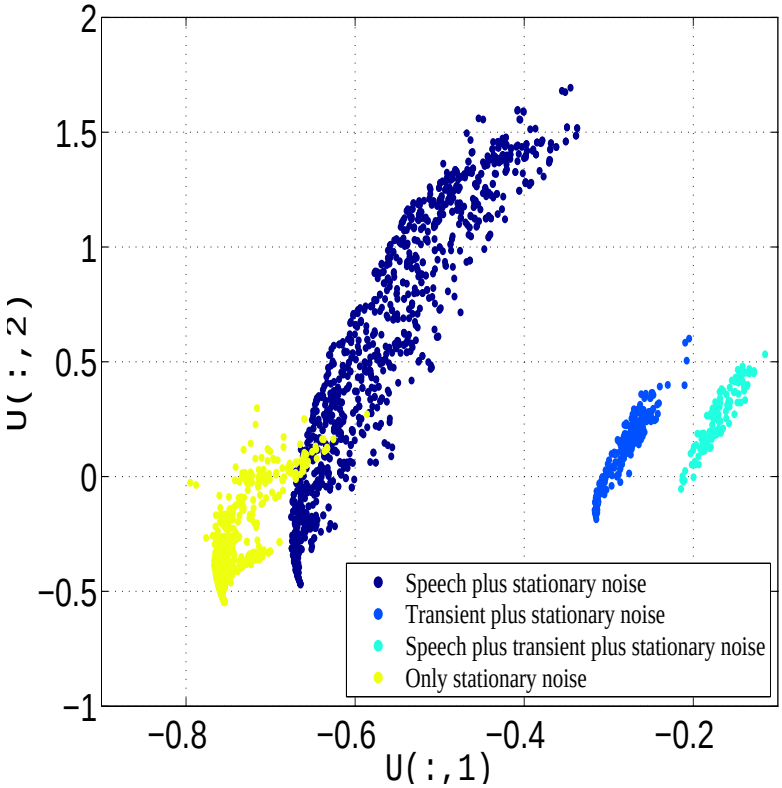


Figure 2.2: Scatter plot of the new representation of a typical training data sequence containing speech and door knock noise degraded by colored Gaussian noise (SNR=5dB) obtained by concatenating all training sequences.

Once the matrix $\boldsymbol{U}$, a new representation of the training data, is obtained, we use Gaussian mixture modeling to model each cluster (i.e. speech presence or absence) with a different GMM. A mixture model is a probabilistic model that assumes the underlying data belongs to a mixture distribution. In a mixture distribution, the density function is a convex combination of other probability density functions. The most common mixture distribution is the Gaussian density function, where each of the mixture components has a Gaussian distribution. This model has been utilized in many machine learning and speech processing applications such as speaker verification [98], texture retrieval [47], and handwriting recognition [18] just to name a few.

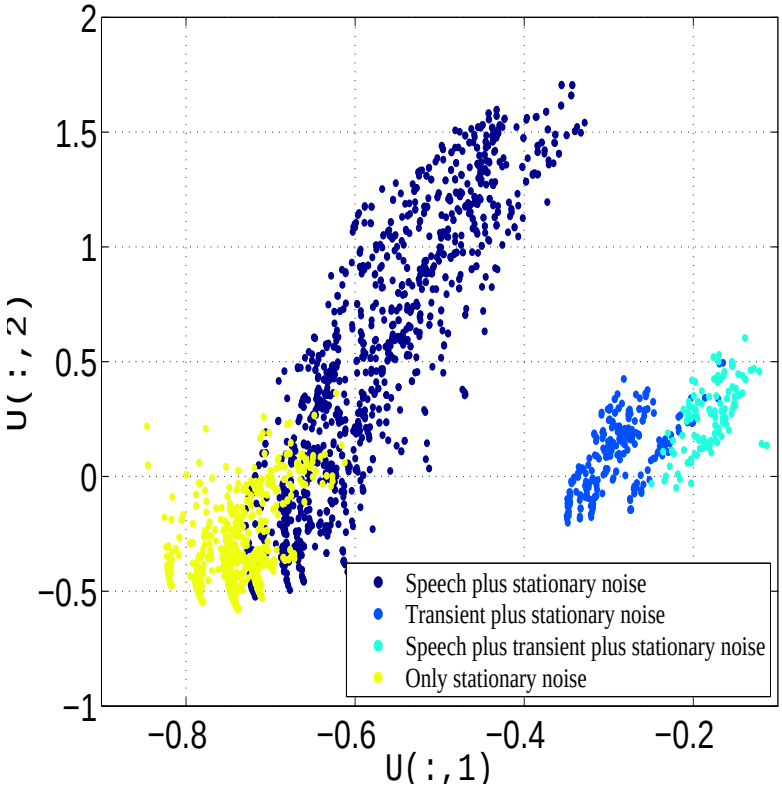


Figure 2.3: Scatter plot of the new representation of a typical training data sequence containing speech and door knock noise degraded by colored Gaussian noise (SNR=5dB) obtained by (2.17)-(2.18).

Parameters of the GMM can be estimated by the Expectation-Maximization (EM) method [17], and the number of Gaussian component to be used can be selected by the Akaike information criterion (AIC) or Bayesian information criterion (BIC). The procedure to obtain the GMM for each cluster is as follows. For each cluster (i.e. speech presence or absence), we find the rows of the matrix $\boldsymbol{U}$ corresponding to that cluster by using the indicator matrix. Then, by exploiting the EM algorithm and AIC or BIC criterion, we fit a GMM to the new data representation in that cluster. Since the matrix $\boldsymbol{U}$ only depends on the training data, the GMM model for each of the two hypotheses (i.e. speech presence or absence) is obtained during the training phase.

Now suppose we are given T frames of unlabeled data, and we want to decide whether each of these frames belongs to the speech presence or speech absence clusters. For each of these frames, we first extract the feature vector using (6.27) and (2.3). Let

$$\boldsymbol{Z}(:,t) = \begin{bmatrix} \boldsymbol{Z}_m(:,t) \\ \Lambda_t^Z \end{bmatrix} \qquad t = 1,2,\cdots,T \tag{2.19}$$

be the feature vector extracted from unlabeled data, where $\boldsymbol{Z}_m(:,t)$ is the absolute value of the

MFCC of the t-th frame, and Λ_t^Z is the likelihood ratio of t-th unlabeled frame obtained by (2.3). The similarity matrix between the new data and training data is computed as follows:

$$\boldsymbol{B} = \left[(\boldsymbol{B}^1_{\boldsymbol{\theta}^{opt}})^T, (\boldsymbol{B}^2_{\boldsymbol{\theta}^{opt}})^T, \cdots, (\boldsymbol{B}^L_{\boldsymbol{\theta}^{opt}})^T\right]^T \tag{2.20}$$

$$\boldsymbol{B}^\ell_{\boldsymbol{\theta}^{opt}}(i,j) = \exp\left(\sum_{p=-P}^{P} -\alpha_p^{opt}\boldsymbol{Q}^\ell(i+p, j+p)\right) \tag{2.21}$$

$$\boldsymbol{Q}^\ell(i,j) = \|\boldsymbol{Y}^\ell_m(:,i)\left(1-\exp(-\Lambda^\ell_i/\varepsilon^{opt})\right) - \boldsymbol{Z}_m(:,j)\left(1-\exp(-\Lambda^Z_j/\varepsilon^{opt})\right)\|_2^2 \tag{2.22}$$

where $\boldsymbol{\theta}^{opt} = [\varepsilon^{opt}, \alpha^{opt}_{-P}, \alpha^{opt}_{-P+1}, \cdots, \alpha^{opt}_{P-1}, \alpha^{opt}_P]$ is the optimum kernel parameters vector obtained in learning stage by solving the optimization problem in (2.15), and $\boldsymbol{B}^\ell_{\boldsymbol{\theta}^{opt}}(i,j)$ $(1 \le i \le N^\ell; 1 \le j \le T)$ is the (i,j)-th element of the matrix $\boldsymbol{B}^\ell_{\boldsymbol{\theta}^{opt}}$. Once the similarity matrix between unlabeled data and training data has been computed, the new data representation in terms of eigenvectors of the Laplacian can be easily approximated by the following equation:

$$\tilde{\boldsymbol{U}} = \mathrm{diag}\left((\boldsymbol{1}\boldsymbol{B}_{k_{nn}})^{-1}\right)\boldsymbol{B}^T_{k_{nn}}\boldsymbol{U}, \tag{2.23}$$

where the i-th column of the matrix $\boldsymbol{B}_{k_{nn}}$ is obtained by setting to zero all elements of the i-th column of $\boldsymbol{B}$, except the k largest elements. The subscript k_{nn} stands for k-nearest neighbor. The last equation means that the low dimensional representation of a given test point is simply the weighted mean of the low representation k-nearest neighbor of that point in the training set. Using this new representation of the unlabeled data, the decision rule can be obtained by a likelihood ratio test as follows. Let $\mathcal{H}_0$ and $\mathcal{H}_1$ be speech absence and presence hypotheses, respectively. Let $f(\cdot;\mathcal{H}_0)$ and $f(\cdot;\mathcal{H}_1)$ be the probability density function of those rows of $\boldsymbol{U}$ corresponding to noise only frames and frames containing speech signal, respectively. These two probability density functions were obtained by GMM modeling in the training stage. The likelihood ratio for a new unlabeled frame is given by:

$$\Gamma_t = \frac{f(\tilde{U}(t,:);\mathcal{H}_1)}{f(\tilde{U}(t,:);\mathcal{H}_0)} \tag{2.24}$$

where $\tilde{U}(t,:)$ is the t-th row of the matrix $\tilde{U}$. Practical evidence shows that using the information supplied by neighboring frames can improve the performance of VAD algorithms [96]. This is because of the fact that frames containing speech signal are usually followed by a frame that also contains speech signal while the transient signals usually last for a single time frame. Using

this fact, the decision rule for an unlabeled time frame is obtained by:

$$\mathrm{VA}_t = \sum_{j=-J}^{J} \Gamma_{t+j} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} T_h \qquad t = 1, 2, \cdots, T \tag{2.25}$$

where T_h is a threshold which controls the tradeoff between probability of detection and false alarm. Increasing (decreasing) this parameter leads to a decrease (increase) of both the probability of false alarm and the probability of detection. In a practical implementation, a hangover scheme is required to lower the probability of false rejections. The hangover scheme does this by reducing the risk of a low-energy portion of speech at the end of an utterance being falsely rejected, by arbitrarily declaring a period of speech activity after a period of speech activity has already been detected. This is based on the idea that speech occurrences are highly correlated with time. We use the hangover technique introduced by Davis et al. in [37]. More specifically, the quantity VA_t is the input of the hangover procedure, and a final VAD decision is obtained from hangover scheme. Our overall voice activity detection algorithm is summarized in Table 2.1.

2.3 Simulation Results and Performance Evaluation

In this section, we examine the performance of the proposed method using several simulations. We also compare the performance of our method with that of conventional statistical model-based methods presented in [21, 61, 96, 108, 111] and two standard VAD's: G.729 [16] and AMR2 [36]. The simulation setup is as follows.

We perform our simulation for different types of stationary and transient noise for different SNR situations. The SNR is defined as the ratio of the speech energy to the energy of stationary noise. The stationary noise energy is computed in those frames that speech signal is present. All speech and transient noise signals are sampled at 16 kHz (although the same performance was obtained at 8 kHz sampling rate) and normalized to have unity as their maximum. Since the duration of transient noise is small with respect to speech, defining SNR for transient noise is not insightful enough. Instead, we normalize the transient noise and speech signal to have the same maximum amplitude, which is a very challenging case to treat [120]. Each signal (speech or transient noise) is approximately 3 sec long. The training and testing sequences are

Table 2.1: Proposed Voice Activity Detection Algorithm Based On Spectral Clustering Method

Learning algorithm:

1. Construct a training data set as described previously that consists of L training signals $\{\boldsymbol{Y}^{\ell} \in \mathrm{R}^{K_m+1\times 3N^{\ell}}; \ell = 1,...,L\}$ together with L indicator matrix $\{\boldsymbol{C}^{\ell} \in \mathrm{R}^{3N^{\ell}\times 4}; \ell = 1,...,L\}$.
2. Solve the optimization problem given in (2.15), to find the optimum value of the parameters (i.e. $\boldsymbol{\theta}^{opt}$).
3. Construct the matrix $\boldsymbol{U}$ by column concatenation of U^1 through U^L by taking into account the sign ambiguity in computation of eigen-vectors (see (2.17) and the text for further discussions).
4. Utilize the EM algorithm to fit a GMM model to the rows of $\boldsymbol{U}$ for each cluster (i.e. speech presence and speech absence).

Output:

$\boldsymbol{U}, f(\cdot;\mathcal{H}_1)$ and $f(\cdot;\mathcal{H}_0)$

Testing Procedure:

Let $z_t(n)$ be the test sequence and $\boldsymbol{Z}_t \in \mathrm{R}^{K_m+1\times N^z}$ the feature matrix extracted from $z(n)$ using (2.19).

for $t = 0 : T : N^z - T$ $(T \ll N^z)$

1. $\boldsymbol{Z} = \boldsymbol{Z}_t(:, t+1 : t+P)$.
2. Compute $\boldsymbol{B}_{k_{nn}}$ using (2.20)-(2.22).
3. Compute the new representation of the unlabeled data using
$$\tilde{\boldsymbol{U}} = \mathrm{diag}\left((\mathbf{1}\boldsymbol{B}_{k_{nn}})^{-1}\right)\boldsymbol{B}_{k_{nn}}^T\boldsymbol{U}.$$
4. Compute the likelihood ratio for a new unlabeled frame by
$$\Gamma_t = \frac{f(\tilde{U}(t,:);\mathcal{H}_1)}{f(\tilde{U}(t,:);\mathcal{H}_0)}.$$
5. The decision rule for an unlabeled time frame is given by
$$\mathrm{VA}_t = \sum_{j=-J}^{J}\Gamma_{t+j} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} T_h \quad .$$
6. Use VA_t and the hangover technique introduced in [37] to obtain the final VAD decision.

end

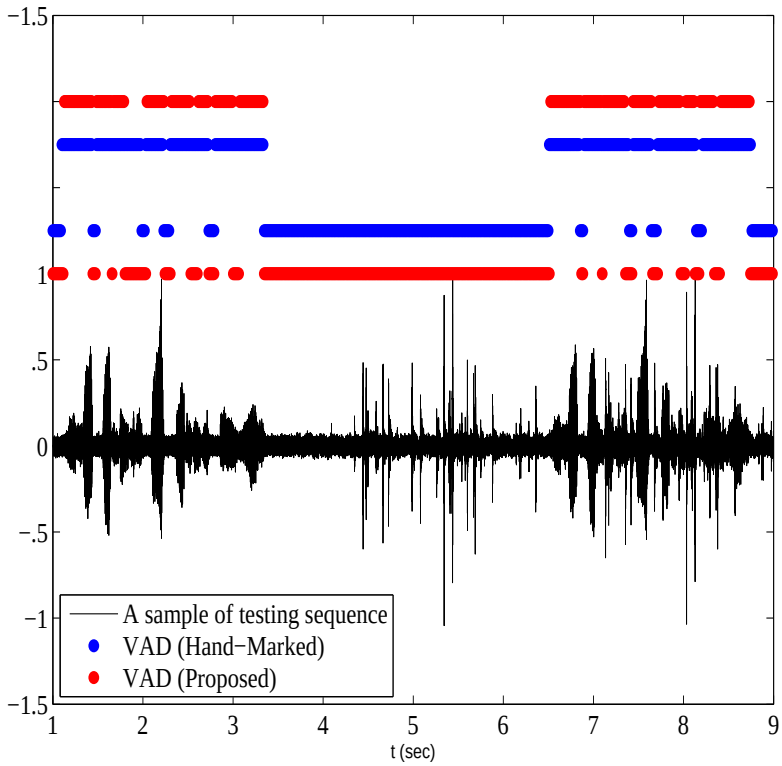


Figure 2.4: A realization of the test signal (speech signal plus keyboard typing plus babble noise) together with hand-marked and proposed VAD.

constructed using the procedure introduced in (2.6) and (2.7). Speech signals are taken from the TIMIT database [50]. Transient noises are taken from [1]. In the training step, we use $M = 50$ different speech utterances (different speakers, half male and half female) and transient noise. In the testing step, we use $M = 50$ different speech utterances (different speakers from the training set, half male and half female) and transient noise (different from training sequences) each approximately 3 sec long (the length of the testing signal is approximately 500 sec, with sixty percent of total frames containing speech). We use windowed STFT with frame length $K_s = 512$ with fifty percent overlap and a hamming window, and compute MFCC in $K_m = 24$ Mel frequency bands. To solve the optimization problem (2.15) in the training stage, we use the $fmincon$ function in Matlab®. We solve this optimization problem under the constraint that all estimated parameters are strictly positive. This constraint results in an appropriate similarity matrix. The parameter k in computing the matrix $\boldsymbol{B}_{k_{nn}}$ (equation (2.23)) is set to 10.

In order to compare our method to the conventional statistical based method, we introduce two different kinds of probability of false alarm. The first type denoted by Pfa, is defined as the

probability that a speech free frame (i.e. consisting of only stationary noise or stationary noise with transient noise) is detected as a speech frame (i.e. exactly the same as probability of false alarm defined in conventional methods). The second type, denoted by $\mathrm{Pfa}_{\mathrm{tr}}$, 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 (which are mostly detected as speech in statistical model-based methods) is small with respect to the total number of frames. Such frames do not affect the probability of false alarm significantly if it is defined as the probability that a noise frame is detected as a speech frame. For our comparison to be fair, we assume that the stationary noise statistics are known in conventional methods. The noise statistics are estimated using a realization of stationary noise. In what follows, we investigate the performance of the proposed method in several situations. For our comparison to be more insightful, we also use the following six parameters defined in [37] to indicate the VAD performance:

- **FEC** (front end clipping): Clipping due to speech being misclassified as noise in passing from noise to speech activity.
- **MSC** (mid speech clipping): Clipping due to speech misclassified as noise during an utterance.
- **BEC** (back end clipping): Clipping due to speech being misclassified as noise in passing from speech activity to noise.
- **Over** (over hang): Noise interpreted as speech due to the VAD flag remaining active in passing from speech activity to noise.
- **NDS** (noise detected as speech): Noise interpreted as speech within a silent period.
- **Correct** (correct VAD decision): Correct decisions made by the VAD.

The results of simulations are depicted in Figures 2.4 through 2.8 and Tables 2.3 through 2.7. Figure 2.4 shows a speech signal corrupted by keyboard typing and babble noise with SNR = 20 dB together with hand-marked VAD and the result of the proposed algorithm. For

this simulation, the detection threshold is chosen such that Pfa_{tr} be zero. The results of Tables 2.3 through 2.7 are obtained by setting the threshold such that the probability of detection (**Correct**) be at least ninety-five percent. In Tables 2.3 through 2.7, the best performance (i.e. the lowest probability of false alarm **NDS**) is identified in bold number. As can be seen from Figures 2.5 through 2.8 and Tables 2.3 through 2.7, although different statistical model-based methods have different performance in different situations, the proposed method has superior performances in all simulations over the compared statistical model-based methods, especially for low false alarm rates. The proposed method outperforms the statistical model-based method even in the case that the training and testing do not match (Figures 2.7 and 2.8 and Tables 2.6 and 2.7). Practical evidence shows that although the choosing appropriate kernel parameters (vector $\boldsymbol{\theta}$) are important in the proposed algorithm, the testing algorithm is not very sensitive to small perturbation of these parameters. This fact can be explained mathematically as the appropriate kernel width lies in an interval and is not just a single point. In particular, for a proper scale, the samples are assumed to lie on a manifold $\mathcal{M}$. It can be shown that the slope of the logarithmic scale of the sum of the kernel weights as a function of the kernel-width, equals to $d/2$, where d is the dimension of the manifold. However, in the limit when kernel-width approaches to infinity, of the sum of the kernel weights equals to N^2, where N is the number of available data. On the other hand, when kernel-width approaches to infinity, of the sum of the kernel weights equals to N. These two limits suggest that the logarithmic plot cannot be linear for all kernel-widths. In the linear region, both the bias and the variance errors are relatively small, and therefore may be chosen from that region (see [35] for more discussion). Using this fact once the optimum parameters are chosen for a specific scenario, it can be used for other situations (not too different) as well, without severely affecting the overall performance of the proposed algorithm. Practical evidence shows that the parameters are relatively sensitive to dramatic change in the maximum of the speech signal. This means the appropriate parameters are chosen (by solving the above-mentioned optimization problem (2.15)) for speech signals with specific maximum, there is no need to estimate them again for change in SNR or the type of stationary or transient noise. Practical evidences show that although the parameters of the kernel are insensitive to change in SNR and stationary or transient noise type, the matrix $\boldsymbol{U}$ is. Hence, for different SNR or stationary or transient noise type, the matrix $\boldsymbol{U}$ must be recomputed (even though simulation results reveal the effectiveness of the proposed method even if there exists a

mismatch between training and testing (Figures 2.7 and 2.8 and Tables 2.5 through 2.7). In all of the following simulations we set the parameter vector to $\boldsymbol{\theta} = .001 \times [300\ \ 0.4\ \ .75\ \ 1\ \ 0.75\ \ 0.4]$. In the testing algorithm we chose $T = 10$, which means that our algorithm has 160 msec delay (each frame is $K_s/(2f_s) = 16$msec long, where the factor of 2 in the denominator is due of fifty percent overlap). Choosing this parameter is a tradeoff between induced delay and computational load. Increasing T leads to a lower computational load but increases the delay. The decrease in computational load by increasing the parameter T is due to the fact that there exist efficient algorithms for computing the matrix B (equation (2.20)), and it can be shown that the computational load is of the order $O(\frac{T+1}{T})$. The most time consuming part of the proposed algorithm is solving the optimization problem, which is done offline and is not of great importance in practice. The excess computational load in the testing stage compared to statistical model-based methods is computation of the matrix $\boldsymbol{B}_{k_{nn}}$. The elapsed time for processing a 500 second sequence sampled at 16kHz for the proposed method and other competing methods are depicted in Table 2.2. Although the computational load of the proposed algorithm is relatively higher than other methods, there exist efficient algorithms for decreasing the computational load [132].

Table 2.2: Elapsed Time in Second for Different Methods.

Proposed	**Sohn**	**Ramirez**	**Chang**	**Shin**	**Ishizuka**	**G729**	**AMR2**
50.41	2.19	2.28	2.30	2.42	4.08	1.51	1.31

Table 2.3: Voice Activity Detection Performance Comparison (**Training**:SNR=10 dB; Stationary Noise: Babble; Transient Noise: Metronome; **Testing**: SNR=10 dB; Stationary Noise: Babble noise; Transient Noise: Metronome).

	Correct	**FEC**	**MSC**	**BEC**	**NDS**	**Over**
Proposed	95.04	0.50	3.37	1.09	**29.25**	1.15
Sohn	99.23	0.37	0.17	0.23	78.06	0.31
Ramirez	98.20	0.26	0.99	0.55	63.95	0.95
Chang	96.09	0.59	1.53	1.79	33.02	0.57
Shin	95.95	0.54	2.47	1.04	68.77	1.44
Ishizuka	95.95	0.54	2.47	1.04	68.77	1.44
G.729	99.94	0.00	0.06	0.00	96.09	0.16
AMR2	99.59	0.24	0.17	0.00	95.4	0.26

Table 2.4: Voice Activity Detection Performance Comparison (**Training**: SNR= 5 dB; Stationary Noise: Babble; Transient Noise: Keyboard Typing; **Testing**: SNR= 5 dB; Stationary Noise: Babble; Transient Noise: Keyboard Typing).

	Correct	**FEC**	**MSC**	**BEC**	**NDS**	**Over**
Proposed	95.16	0.61	3.46	0.77	**53.44**	1.46
Sohn	98.79	0.36	0.52	0.32	82.00	0.25
Ramirez	97.90	0.42	1.20	0.47	68.55	0.89
Chang	95.71	0.81	2.22	1.26	63.18	0.76
Shin	95.51	0.64	2.52	1.33	66.11	0.88
Ishizuka	95.65	0.83	2.51	1.01	59.23	1.43
G.729	99.95	0.05	0.00	0.00	96.85	0.22
AMR2	99.43	0.40	0.16	0.97	96.03	0.85

Table 2.5: Voice Activity Detection Performance Comparison (**Training**: SNR= 5 dB; Stationary Noise: Babble; Transient Noise: Keyboard Typing; **Testing**: SNR= 5 dB; Stationary Noise: White Gaussian; Transient Noise: Metronome) .

	Correct	**FEC**	**MSC**	**BEC**	**NDS**	**Over**
Proposed	95.05	0.95	3.04	0.95	**45.39**	2.16
Sohn	96.35	0.90	1.59	1.17	87.12	0.66
Ramirez	95.94	0.60	2.69	0.77	83.53	1.11
Chang	95.97	0.88	2.06	1.09	77.79	0.81
Shin	95.79	0.85	1.67	1.69	70.56	0.83
Ishizuka	96.11	0.65	1.88	1.36	62.51	1.67
G.729	99.91	0.04	0.02	0.02	95.33	0.16
AMR2	98.96	0.07	0.96	0.88	94.82	0.33

Table 2.6: Voice Activity Detection Performance Comparison (**Training**: SNR= 10 dB; Stationary Noise: Colored Gaussian; Transient Noise: Door Knock; **Testing**: SNR= 10 dB; Stationary Noise: Babble; Transient Noise: Metronome) .

	Correct	**FEC**	**MSC**	**BEC**	**NDS**	**Over**
Proposed	95.03	0.55	3.01	1.42	**24.04**	1.40
Sohn	99.23	0.37	0.17	0.23	78.06	0.31
Ramirez	98.20	0.26	0.99	0.55	63.95	0.95
Chang	96.09	0.59	1.53	1.79	33.02	0.57
Shin	95.85	0.61	1.70	1.84	45.04	0.88
Ishizuka	95.95	0.54	2.47	1.04	68.77	1.44
G.729	99.94	0.00	0.06	0.00	96.09	0.16
AMR2	99.05	0.94	0.01	0.06	95.95	0.62

Table 2.7: Voice Activity Detection Performance Comparison (**Training**: SNR= 20 dB; Stationary Noise: Babble; Transient Noise: Door Knock; **Testing**: SNR= 5 dB; Stationary Noise: Colored Gaussian; Transient Noise: Keyboard typing) .

	Correct	**FEC**	**MSC**	**BEC**	**NDS**	**Over**
Proposed	95.03	0.88	2.83	1.26	**31.47**	1.29
Sohn	98.01	0.37	0.79	0.83	76.01	0.86
Ramirez	97.23	0.45	1.75	0.57	74.28	0.98
Chang	96.29	0.59	1.52	1.60	60.84	0.76
Shin	95.68	0.43	2.68	1.20	66.51	0.82
Ishizuka	95.82	0.57	2.34	1.27	22.13	1.57
G.729	99.98	0.00	0.02	0.00	96.57	0.16
AMR2	99.51	0.49	0.10	0.94	95.93	0.21

2.4 Conclusions

In this chapter, we have proposed a novel voice activity detector based on spectral clustering method. Our main concern was dealing with transient noise, which is very challenging to handle. Almost all conventional methods fail in this situation. Our VAD is a supervised learning algorithm that requires some training data in order to estimate the parameters of the kernel used for computation of a similarity matrix. We used GMM to model the eigenvectors of the similarity matrix. In the testing stage, we used eigenvector extension and proposed a VAD which can be used for online processing of the data with a small delay. Simulation results have demonstrated the high performance of the proposed method, particularly its advantage in treating transient noises.

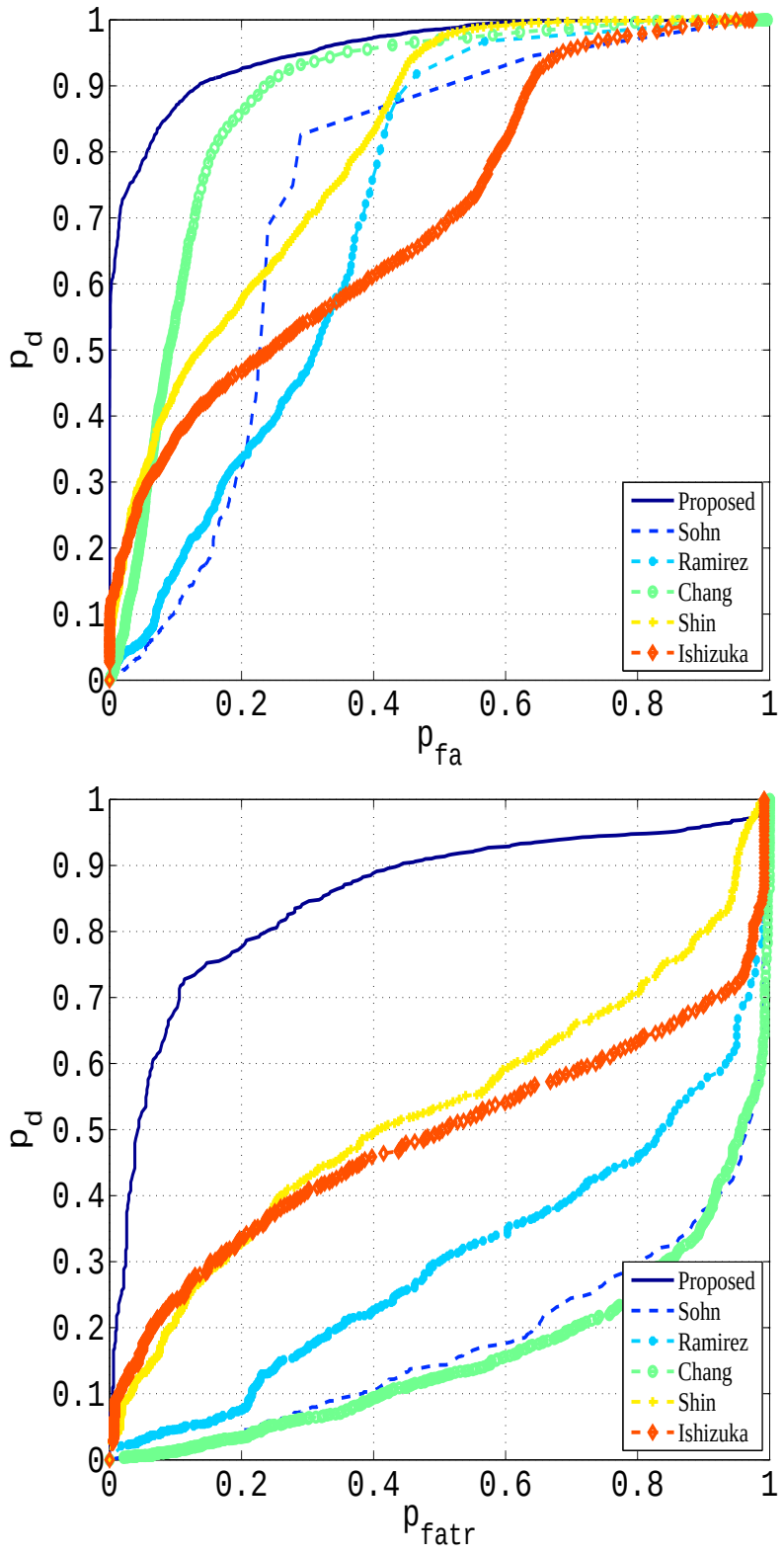


Figure 2.5: Probability of detection versus probability of false alarm (**Training**: SNR=10 dB; Stationary Noise: Babble; Transient Noise: Metronome; **Testing**: SNR=10 dB; Stationary Noise: Babble; Transient Noise: Metronome).

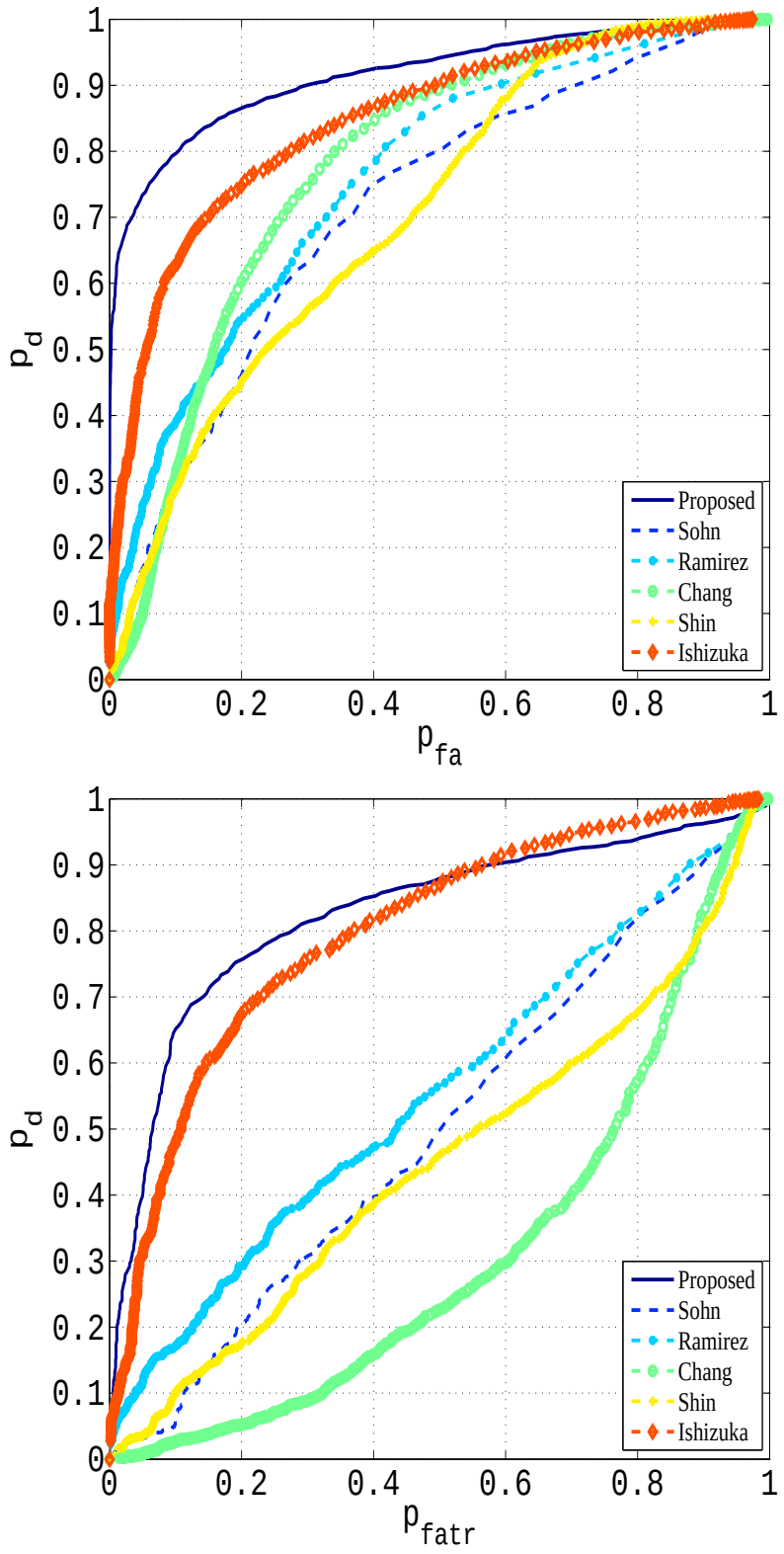


Figure 2.6: Probability of detection versus probability of false alarm (**Training**: SNR= 5 dB, Stationary Noise: Babble; Transient Noise: Keyboard Typing; **Testing**: SNR= 5 dB; Stationary Noise: Babble; Transient Noise: Keyboard Typing).

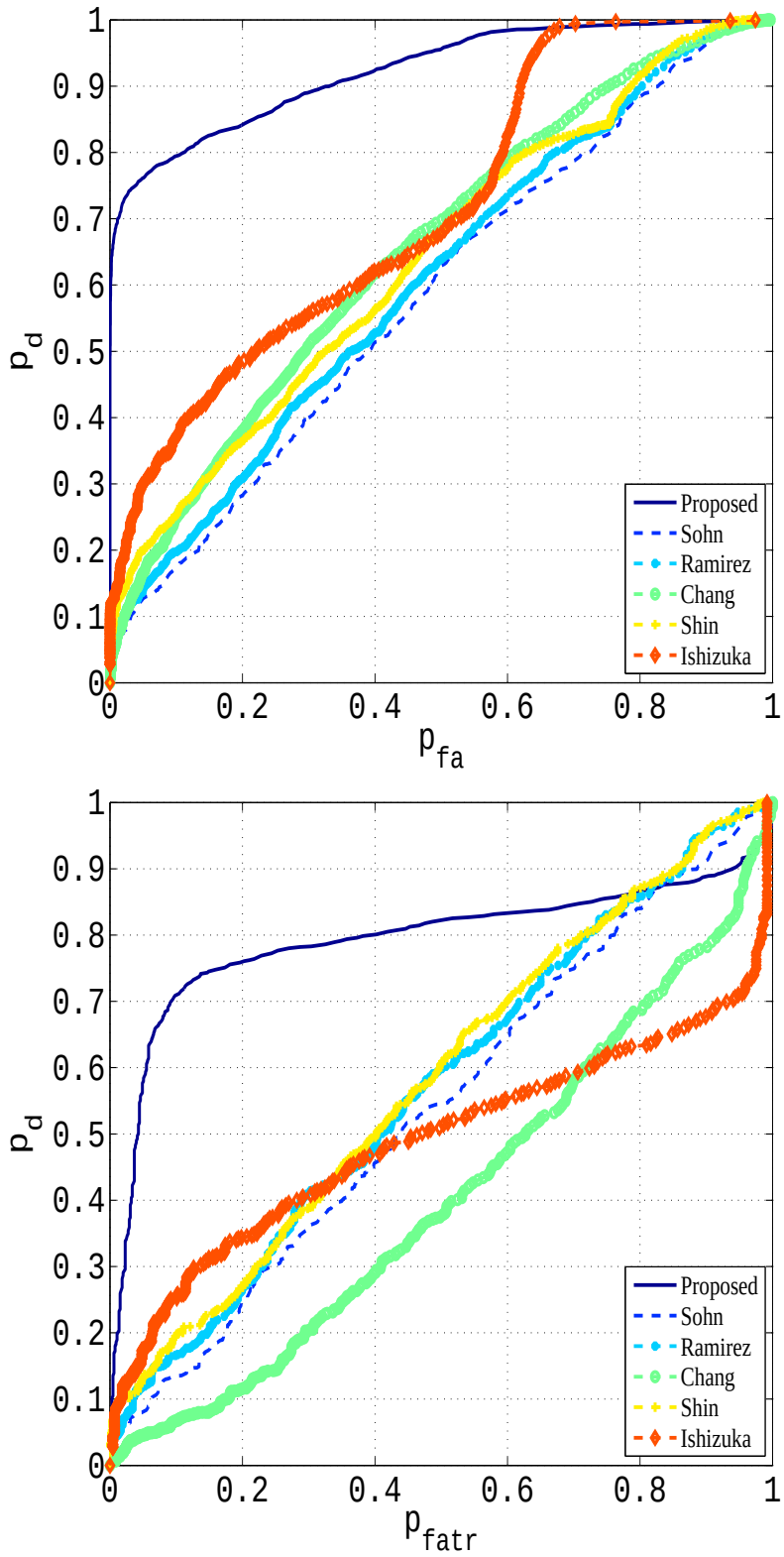


Figure 2.7: Probability of detection versus probability of false alarm (**Training**: SNR= 5 dB; Stationary Noise: Babble; Transient Noise: Keyboard Typing; **Testing**: SNR= 5 dB; Stationary Noise: White Gaussian; Transient Noise: Metronome) .

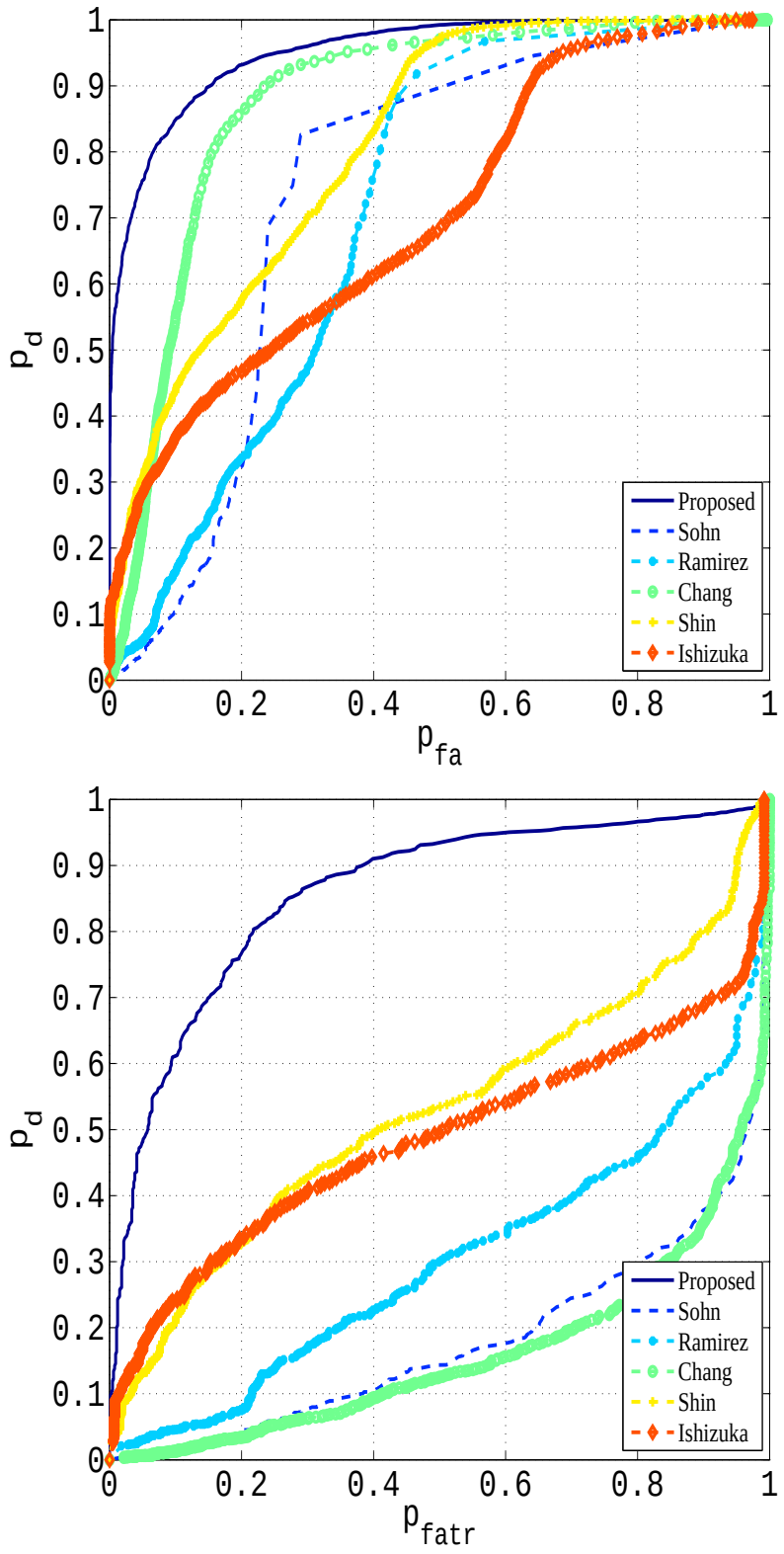


Figure 2.8: Probability of detection versus probability of false alarm (**Training**: SNR= 10 dB; Stationary Noise: Colored Gaussian; Transient Noise: Door Knock; **Testing**: SNR= 10 dB; Stationary Noise: Babble; Transient Noise: Metronome) .

Chapter 3

Embedding and Function Extension on Directed Graph

3.1 Introduction

Recent advances in geometric based methods of data mining and machine learning lead to efficient algorithms for lots of applications such as dimensionality reduction, function extension, classification and clustering, just to name a few. Most of these methods are graph based techniques. Graphs offer an advantageous compromise between their simplicity, interpretability and their ability to express complex relationships between data points. The core idea in such algorithms is to construct a weighted graph on data points such that each vertex of the graph represents a data point, and a weighted edge, connecting two vertices to each other, represents the similarity between the two corresponding data points. In the context of networks (e.g., social networks), the data naturally lead themselves to graph modelling [59]. The graph based representation of data combined with Markov chain techniques exhibits extremely successful results. The main idea here is based on the fact that the eigenvectors of Markov matrices can be regarded as coordinates on the data set. Among vast techniques incorporating Markov chain methods in data processing, kernel eigenmap methods have attracted much research attention recently. The algorithmic consequence of these methods are local linear embedding (LLE) [99], Laplacian eigenmaps [14], Hessian eigenmaps [40], local tangent space alignment [134] and diffusion maps [32].

In most of these kernel eigenmaps based methods, the similarity between points is given by

a symmetric positive semi-definite kernel. In some practical applications the similarity between points is not necessarily symmetric. Typical examples are web information retrieval based on hyperlink structure, document classification based on citation graphs [53], web information retrieval based on hyperlink structure, and protein clustering based on pair-wise alignment scores [92]. Some works have been done to deal with the ranking problem on link structure of the Web. Although much progress in the field, it is still a hard task to do general data analysis on directed graphs such as classification and clustering. Chen et al. [22], proposed an algorithm for embedding vertices on directed graphs to vector spaces. This algorithm explores the inherent pair-wise relation between vertices of the directed graph by using transition probability and the stationary distribution of Markov random walks, and embeds the vertices into vector spaces preserving such relation optimally. Recently, Perrault-Joncas and Meilă [93] proposed an algorithm based on analysis of Laplacian type operators and their continuous limit as generators of diffusions on a manifold. They modelled the observed graph as a sample from a manifold endowed with a vector field, and designed an algorithm that separates and recovers the features of this process: the geometry of the manifold, the data density, and the vector field. The most important shortcoming of these methods is not providing a straightforward procedure to extend the embedding to new points in case only the distances of the new point to the original data points are known, which encountered in applications. In [31], Coifman and Hirn introduced a simple procedure for construction of a bi-stochastic kernel for an arbitrary data set that is derived from an asymmetric affinity function. The affinity function measures the similarity between points in test set and some reference set.

These geometrically based algorithms have been applied in various signal processing applications. In a pioneering work, Shi and Malik [107] used spectral methods for image segmentation. Later many researchers have used geometrically based methods in applications such as image clustering [46, 133], image completion [52], speech enhancement in presence of transient noise [120], voice activity detection in presence of transient noise [85], linear and nonlinear independent component analysis [109, 110], parametrization of linear systems [121], and single channel source localization [119]. Most if not all, of these applications can be regarded as out of sample function extension, in which an empirical function is extended to unlabelled data. In these applications, usually a large amount of data is involved and the only way to perform a task like clustering, regression, or classification is to sub-sample the data set $\overline{X}$ in order to

reduce the size of the problem, process the new set X, and then extend the results to the original data $\overline{X}$. Coifman and Lafon [27], proposed a geometric harmonics procedure, inspired from the Nyström method, to perform this task. More specifically, they assumed that the similarity between the data points is given by a symmetric positive semi-definite kernel. Then it is shown that the eigenfunctions of the integration operator defined by this kernel, form an orthogonal basis for the space of squared integrable functions defined on $\overline{X}$ (i.e. $L^2(\overline{X})$). In order to extend a function defined on the set X to the data set $\overline{X}$, first the eigenfunctions computed on X are extended to the data set $\overline{X}$ using the Nyström method. The function is then approximated as the weighted sum of these extended eigenfunctions. Kushnir et al. [70] and Singer et al. [110] introduced a method for parameterizing high dimensional data into its independent physical parameters, which enables the identication of the parameters and a supervised extension of the re-parametrization to new observations. In their work, a novel diffusion processes was used, utilizing only the small observed set, that approximates the isotropic diffusion on the parametric manifold. They utilized Nyström-type extension of the embedding of that small observed data-set to the embedding into the independent components on a much larger data-set.

In this chapter, we propose a novel technique for embedding a directed graph to Euclidean space. We model the observed data as samples from a manifold where the similarity between the points is given by an asymmetric kernel. This asymmetric kernel is modelled utilizing a vector field, and we design an algorithm that separates and recovers the geometry of the manifold, the data density, and the vector field. We further provide a simple Nyström extension procedure which allows us to extend both the embedding and the estimated vector field to new data points. More precisely, we adopt the method presented in [70] into the case when the kernel is asymmetric. The rest of this chapter is organized as follows. In Section 3.2, we provide a model which can be used in directed graph modelling. We also introduce our results regarding the limit of Laplacian type operators and provide an algorithm for obtaining the embedding of a directed graph. We also propose a Nyström extension procedure for extending the embedding and the vector field to new data points. In Section 3.3, we provide some experimental results. We conclude the chapter in Section 3.4.

3.2 Problem Formulation, Embedding and Function Extension

Let X be a set of n data points sampled according to a distribution $p = e^{-U}$ from an unknown smooth manifold $\mathcal{M} \subset \mathrm{R}^{\ell}$ with intrinsic dimension $d < \ell$. Let G be a directed graph with n nodes constructed from the data set X, where each nodes of the graph (e.g. the node i) corresponds to a point in the set X (e.g. $x_i \in X$). We assume that the edge weight $K_{i,j}$ between nodes i and j, is given by a positive asymmetric similarity kernel $k_{\varepsilon}(\cdot,\cdot)$ (i.e. $K_{i,j} = k_{\varepsilon}(x_i, x_j) \geq 0$). We also assume that the directional component of $k_{\varepsilon}(\cdot,\cdot)$ is derived by a vector field $\boldsymbol{r}$ on the manifold $\mathcal{M}$, which will be precisely defined shortly. This vector field $\boldsymbol{r}$, is sufficient to characterize any directionality associated with a drift component and as it turns out, the component of $\boldsymbol{r}$ normal to $\mathcal{M} \subset \mathrm{R}^{\ell}$ can also be used to characterize any source component, see [93] for further discussion. The problem is finding an embedding of G into $\mathrm{R}^m; m \leq d$ which approximates the generative processs geometry $\mathcal{M}$, the sampling distribution $p = e^{-U}$, and the directionality $\boldsymbol{r}$. This embedding needs to be consistent as sample size increases and the bandwidth of the kernel vanishes.

3.2.1 Anisotropic Diffusion Operator

Any kernel $k_{\varepsilon}(x,y)$ can be decomposed into symmetric and anti-symmetric parts as follows

$$k_{\varepsilon}(x,y) = h_{\varepsilon}(x,y) + a_{\varepsilon}(x,y), \tag{3.1}$$

where $h_{\varepsilon}(x,y) = h_{\varepsilon}(y,x)$ is the symmetric component and $a_{\varepsilon}(x,y) = -a_{\varepsilon}(y,x)$ is the antisymmetric component of the kernel. As in [93], we assume that the symmetric and anti-symmetric part can be written as

$$h_{\varepsilon}(x,y) = \frac{h(\|x-y\|^2/\varepsilon)}{\varepsilon^{d/2}} \tag{3.2}$$

$$a_{\varepsilon}(x,y) = \frac{\boldsymbol{r}(x,y)}{2} \cdot (y-x) \frac{h(\|x-y\|^2/\varepsilon)}{\varepsilon^{d/2}}, \tag{3.3}$$

respectively, where $\boldsymbol{r}(x,y) = \boldsymbol{r}(y,x)$ and $h \geq 0$ is an arbitrary exponentially decreasing function when $\|x-y\|$ converges to infinity.

In [93] an embedding algorithm has been proposed based on symmetrization of this kernel. More specifically, assuming that $k_{\varepsilon}(x,y)$ is the asymmetric similarity kernel, the authors defined

a symmetric kernel $s_\varepsilon(x,y) : \mathcal{M} \times \mathcal{M} \to \mathrm{R}^+$ as follows

$$s_\varepsilon(x,y) = s_\varepsilon(y,x) = \frac{k_\varepsilon(x,y) + k_\varepsilon(y,x)}{2}. \tag{3.4}$$

Then the diffusion map [32] has been applied to this symmetric kernel in order to find the graph embedding. The vector field $\boldsymbol{r}$ and the sampling density p are then obtained by constructing a new diffusion operator using $k_\varepsilon(\cdot,\cdot)$. Although this method succeeds in obtaining the embedding and estimating the sampling distribution and the vector field, the method suffers from a serious limitation. Suppose that for a set of data $X = \{x_1, x_2, \cdots, x_n\}$ the embedding is found by performing the above mentioned procedure. Now we want to approximate the embedding of a new data point y using Nyström like eigenfunction extension. Since the kernel is asymmetric, we need to know both the distance (i.e. the affinity measure) of the new point y from all of the points in X (i.e. $k_\varepsilon(y,x_i); 1 \le i \le n$) and the distances of all of the points in X from the new point y (i.e. $k_\varepsilon(x_i,y); 1 \le i \le n$). In practical applications, one of these sets of distances might be unknown which restricts this method.

Here, we solve the above mentioned problem by choosing a different symmetrization of the affinity kernel. Using the asymmetric kernel $k_\varepsilon(x,y)$, we define a symmetric kernel $s_\varepsilon(x,y) : \mathcal{M} \times \mathcal{M} \to \mathrm{R}^+$ as follows

$$s_\varepsilon(x,y) = s_\varepsilon(y,x) = \int_{\mathcal{M}} k_\varepsilon(x,t) k_\varepsilon(y,t) dt. \tag{3.5}$$

Note that this kernel is positive semi-definite, i.e., for any $m \ge 1$ and any choice of real numbers $\alpha_1, \cdots, \alpha_m$, and points $x_1, \cdots, x_m$ in $\mathcal{M}$, we have

$$\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j s_\varepsilon(x_i,x_j) = \sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j \int_{\mathcal{M}} k_\varepsilon(x_i,t) k_\varepsilon(x_j,t) dt = \int_{\mathcal{M}} \left(\sum_{i=1}^{m} \alpha_i k_\varepsilon(x_i,t) \right)^2 dt \ge 0. \tag{3.6}$$

These two kernels, i.e. the asymmetric kernel $k_\varepsilon(x,y)$ and its symmetrized version $s_\varepsilon(x,y)$, can be used to construct a family of Laplacian type operators suitable for graph embedding as in [32, 34, 125]. This family of operators is inspired by anisotropic diffusion operator introduced in [32] for embedding undirected graph. We define a family of diffusion operators as follows. Let $\alpha \in [0,1]$ and

$$p_{k,\varepsilon}(x) = \int_{\mathcal{M}} k_\varepsilon(x,y) p(y) dy \tag{3.7}$$

$$p_{s,\varepsilon}(x) = \int_{\mathcal{M}} s_\varepsilon(x,y) p(y) dy. \tag{3.8}$$

We form a new kernel by

$$k_\varepsilon^{(\alpha)}(x,y) = \frac{k_\varepsilon(x,y)}{p_{k,\varepsilon}^\alpha(x)p_{k,\varepsilon}^\alpha(y)} \tag{3.9}$$

$$s_\varepsilon^{(\alpha)}(x,y) = \frac{s_\varepsilon(x,y)}{p_{s,\varepsilon}^\alpha(x)p_{s,\varepsilon}^\alpha(y)}. \tag{3.10}$$

Then, we apply the weighted graph Laplacian normalization to these kernels by setting the outdegree distributions of the newly defined kernels as

$$d_{k,\varepsilon}(x) = \int_{\mathcal{M}} k_\varepsilon^{(\alpha)}(x,y)p(y)dy \tag{3.11}$$

$$d_{s,\varepsilon}(x) = \int_{\mathcal{M}} s_\varepsilon^{(\alpha)}(x,y)p(y)dy, \tag{3.12}$$

and by defining the anisotropic transition kernel by

$$t_{k,\varepsilon,\alpha}(x,y) = \frac{k_\varepsilon^{(\alpha)}(x,y)}{d_{k,\varepsilon}(x)} \tag{3.13}$$

$$t_{s,\varepsilon,\alpha}(x,y) = \frac{s_\varepsilon^{(\alpha)}(x,y)}{d_{s,\varepsilon}(x)}. \tag{3.14}$$

Finally we define our diffusion operator as follows

$$T_{\varepsilon,\alpha}^{aa} f(x) = \int_{\mathcal{M}} t_{k,\varepsilon,\alpha}(x,y)f(y)p(y)dy \tag{3.15}$$

$$T_{\varepsilon,\alpha}^{ss} f(x) = \int_{\mathcal{M}} t_{s,\varepsilon,\alpha}(x,y)f(y)p(y)dy, \tag{3.16}$$

where the superscripts a and s indicates which kernel (asymmetric $k_\varepsilon(x,y)$ or symmetric $s_\varepsilon(x,y)$) is used for obtaining the anisotropic kernel and the out-degree distribution. It is worth mentioning that there are in fact eight possible combinations of kernel and degree distribution. Since only the two above-mentioned diffusion operators are of particular interest in retrieving the generative processs geometry $\mathcal{M}$, the sampling distribution $p = e^{-U}$, and the directionality $\boldsymbol{r}$, we concentrate on obtaining the infinitesimal generators of these diffusion operators as ε converges to 0.

3.2.2 Continuous Limit of the Diffusion Operator

In order to find the infinitesimal generators of the diffusion operators we need the following two lemmas.

Lemma 3.2.1. *Let $\mathcal{M}$ be a compact, closed, smooth manifold of dimension d and $k_\varepsilon(x,y)$ be an asymmetric similarity kernel defined by equations (3.1)-(3.3), then for any function $f \in C^3(\mathcal{M})$, the integral operator based on k_ε has the asymptotic expansion*

$$\int_{\mathcal{M}} k_\varepsilon(x,y)f(y)dy = m_0 f(x) + \varepsilon\frac{m_2}{2}\left(\omega(x)f(x) + \triangle f(x) + \boldsymbol{r}\cdot\nabla f(x) + f(x)\nabla\cdot\boldsymbol{r}\right) + O(\varepsilon^2) \tag{3.17}$$

and

$$\int_{\mathcal{M}} k_\varepsilon(y,x)f(y)dy = m_0 f(x) + \varepsilon\frac{m_2}{2}\left(\omega(x)f(x) + \triangle f(x) - \boldsymbol{r}\cdot\nabla f(x) - f(x)\nabla\cdot\boldsymbol{r}\right) + O(\varepsilon^2), \tag{3.18}$$

where $m_0 = \int_{\mathrm{R}^d} h(\|u\|^2)du$ and $m_2 = \int_{\mathrm{R}^d} u_1^2 h(\|u\|^2)du$, u_1 is the first element of a vector u and $\omega(x) = \frac{2}{m_2}\int_{u\in\mathrm{R}^d}\left(Q_2(u)h\left(\|u\|^2\right) + Q_4(u)h'\left(\|u\|^2\right)\right)du$.

Proof. See Appendix A. □

Lemma 3.2.2. *Suppose the conditions of Lemma 3.2.1 are satisfied, then*

$$\int_{\mathcal{M}} s_\varepsilon(x,y)f(y)dy = \int_{\mathcal{M}}\int_{\mathcal{M}} k_\varepsilon(x,t)k_\varepsilon(y,t)dt f(y)dy = m_0^2 f(x) + \varepsilon m_0 m_2\left(\omega(x)f(x) + \triangle f(x)\right) + O(\varepsilon^2). \tag{3.19}$$

Proof. Let

$$\begin{aligned} G_\varepsilon(x) &= \int_{\mathcal{M}}\int_{\mathcal{M}} k_\varepsilon(x,t)k_\varepsilon(y,t)f(y)dydt \\ &= \int_{\mathcal{M}} k_\varepsilon(x,t)\left(\int_{\mathcal{M}} k_\varepsilon(y,t)f(y)dy\right)dt. \end{aligned} \tag{3.20}$$

Using Lemma 3.2.1 (equation (3.18)) we have

$$G_\varepsilon(x) = \int_{\mathcal{M}} k_\varepsilon(x,t)\left(m_0 f(t) + \varepsilon\frac{m_2}{2}g(t)\right)dt + O(\varepsilon^2) \tag{3.21}$$

where

$$g(x) = \omega(x)f(x) + \triangle f(x) - \boldsymbol{r}\cdot\nabla f(x) - f(x)\nabla\cdot\boldsymbol{r}. \tag{3.22}$$

Now applying Lemma 3.2.1 again to equation (3.21) we have

$$\begin{aligned} G_\varepsilon(x) = m_0\left[m_0 f(x) + \varepsilon\frac{m_2}{2}g(x)\right] + \varepsilon\frac{m_2}{2}\Big[&\omega(x)\left(m_0 f(x) + \varepsilon\frac{m_2}{2}g(x)\right) + \triangle\left(m_0 f(x) + \varepsilon\frac{m_2}{2}g(x)\right) \\ &+ \boldsymbol{r}\cdot\nabla\left(m_0 f(x) + \varepsilon\frac{m_2}{2}g(x)\right) + \nabla\cdot\boldsymbol{r}\left(m_0 f(x) + \varepsilon\frac{m_2}{2}g(x)\right)\Big] + O(\varepsilon^2). \end{aligned} \tag{3.23}$$

Discarding the second order terms we get

$$\begin{aligned} G_\varepsilon(t) &= m_0^2 f(t) + \varepsilon \frac{m_2 m_0}{2} g(t) + \varepsilon \frac{m_2}{2} \left[\omega(x) m_0 f(t) + \triangle\left(m_0 f(t)\right) + \boldsymbol{r} \cdot \nabla\left(m_0 f(t)\right) + \nabla \cdot \boldsymbol{r}\left(m_0 f(t)\right)\right] + O(\varepsilon^2) \\ &= m_0^2 f(t) + \varepsilon \frac{m_2 m_0}{2} \left[g(t) + \omega(x) f(x) + \triangle f(t) + \boldsymbol{r} \cdot \nabla f(t) + \nabla \cdot \boldsymbol{r} f(t)\right] + O(\varepsilon^2). \end{aligned} \tag{3.24}$$

Using the definition of $g(x)$ in (3.22) we get the final result, i.e.

$$G_\varepsilon(t) = m_0^2 f(t) + \varepsilon m_2 m_0 \left[\omega(x) f(x) + \triangle f(t)\right] + O(\varepsilon^2). \tag{3.25}$$

□

The scalar functions $\omega(x)$ was first obtained in [32] and corresponds to an interaction between the symmetric kernel $h_\varepsilon(\cdot,\cdot)$ and the curvature of $\mathcal{M}$. Lemma 1 gives a general fact about spectral embedding algorithms: In most cases, Laplacian operators confound the effects of spatial proximity, sampling density and directional flow due to the presence of the various terms above.

The next theorem, represents the main result of this chapter regarding the infinitesimal generator of the diffusions introduced before.

Theorem 3.2.3. *Suppose that the kernel $k(x,y)$ is normalized such that $m_0 = 1$ and $m_2 = 2$. Let*

$$L^{aa}_{\varepsilon,\alpha} = \frac{I - T^{aa}_{\varepsilon,\alpha}}{\varepsilon} \tag{3.26}$$

$$L^{ss}_{\varepsilon,\alpha} = \frac{I - T^{ss}_{\varepsilon,\alpha}}{2\varepsilon} \tag{3.27}$$

be the infinitesimal generator of the Markov chain, where $T^{aa}_{\varepsilon,\alpha}$ and $T^{ss}_{\varepsilon,\alpha}$ are defined in (3.15) and (3.16) , respectively. Then

$$\lim_{\varepsilon \to 0} L^{aa}_{\varepsilon,\alpha} f = \triangle f - 2(1-\alpha)\nabla U \cdot \nabla f + \boldsymbol{r} \cdot \nabla f \tag{3.28}$$

$$\lim_{\varepsilon \to 0} L^{ss}_{\varepsilon,\alpha} f = \triangle f - 2(1-\alpha)\nabla U \cdot \nabla f \tag{3.29}$$

Proof. See Appendix B. □

In the next section, we show how this theorem can be used to recover the geometry of data along with the vector field.

3.2.3 Manifold Learning, Estimation of Sampling Density and the Vector Field

The geometry of the data can be fully recovered by computation of the Laplace-Beltrami operator on the manifold. In other words, the geometry of a Riemannian manifold is determined by the spectrum of Laplace-Beltrami operator. Additionally, since the eigenfunctions of this operator make a basis for all functions in $L^2(\mathcal{M})$, obtaining the eigenfunctions of this operator plays an important role in manifold learning. In order to fully learn the manifold $\mathcal{M}$, we also need to estimate the sampling density $p = e^{-U}$ and the vector field $\boldsymbol{r}$. As shown in [29, 93] the eigenfunctions of the Laplace-Beltrami operator can be estimated by computing the eigenfunctions of $L^{ss}_{\varepsilon,1}$. Furthermore, the sampling density can be obtained by computing the left eigenfunctions of the $L^{ss}_{\varepsilon,1}$ corresponding to eigenvalue 1.

Since

$$\lim_{\varepsilon\to 0} L^{aa}_{\varepsilon,\alpha} f - L^{ss}_{\varepsilon,1} f = \boldsymbol{r}\cdot\nabla f, \tag{3.30}$$

the components of the vector field $\boldsymbol{r}$ on the tangent space denoted by $\boldsymbol{r}_{||}$, can be recovered by

$$\boldsymbol{r}_{||} = \boldsymbol{r}\cdot\nabla\psi, \tag{3.31}$$

where ψ is any diffeomorphic embedding of $\mathcal{M}$. Because of the fact that the eigenfunctions of the Laplace-Beltrami operator is a diffeomorphic embedding of $\mathcal{M}$, we use the eigenfunctions of this operator in (3.31) to recover the vector field $\boldsymbol{r}$. The algorithmic consequence of this procedure for learning the manifold using a directed kernel is summarized in Table 3.2.

3.2.4 Embedding and Function Extension on Directed Graph

In this section, we present a method for embedding and function extension on directed graph which essentially originates from Nyström eigenfunction extension. We adopt the method presented in [70] into the case when the kernel is asymmetric. More specifically, suppose that we have a data set $X \subset \mathcal{M}$ consisting of n data points. Let $\psi(X) \in \mathrm{R}^{n\times m}$ be the embedding of these points into R^m obtained using the Algorithm 3.2. Furthermore, suppose that we are given a function $f(\cdot)$ whose value is known only on the set X. Now, assume that we are given another set of data points $\overline{X} \subset \mathcal{M}$ and the of distances $\{k_\varepsilon(\bar{x},x); \bar{x}\in\overline{X}, x\in X\}$. The goal is extending the embedding and finding the value of the function $f(\cdot)$ on this set $\overline{X} \subset \mathcal{M}$.

As shown earlier in (3.6), $s_\varepsilon(x,y)$ the symmetrized version of the asymmetric kernel $k_\varepsilon(x,y)$, is positive semi-definite. Using $s_\varepsilon(x,y)$ we define an integral operator by

$$\boldsymbol{S}f(x) = \int_X s_\varepsilon(x,y)f(y)dy. \tag{3.32}$$

It is trivial to show that this operator is self-adjoint and compact and hence has a discrete sequence of eigenvalues $\{\lambda_j\}$ (in nonincreasing order) and eigenvectors $\{\psi_j\}$ defined on X where

$$\lambda_j\psi_j(x) = \int_X s_\varepsilon(x,y)\psi_j(y)dy, \tag{3.33}$$

and since the operator is positive semi-definite, $\lambda_j \geq 0; \forall j$. The fact that the kernel can be evaluated on the entire $\mathcal{M}$, enables us to take any $\bar{x} \in \mathcal{M}$ in the right-hand side of this identity. This yields the following Nyström extension of ψ_j from X to $\mathcal{M}$, i.e.

$$\overline{\psi}_j(\bar{x}) = \frac{1}{\lambda_j}\int_X s_\varepsilon(\bar{x},y)\psi_j(y)dy; \qquad \bar{x} \in \mathcal{M}, \tag{3.34}$$

where $\overline{\psi}_j(\bar{x})$ is the *extended* value of the eigenvector $\psi_j(\cdot)$ at the point $\bar{x}$. Note that since

$$s_\varepsilon(\bar{x},y) = \int_{\mathcal{M}} k_\varepsilon(\bar{x},t)k_\varepsilon(y,t)dt, \tag{3.35}$$

this eigenfunction extension technique only needs the distance from the test point $\bar{x}$ to the training points $y \in \mathcal{M}$. Hence, in cases where the distance from the training points to the test point $\bar{x}$ is not available, this method outperforms the method presented in [93].

Our function extension algorithm relies on the fact that the Laplace-Beltrami operator is a positive semi-definite operator and therefore its eigenfunctions form a basis for the set of functions defined on the manifold. In particular, since the eigenfunctions $\psi_j(x); x \in X$ form a basis for the set of functions on X, any function $f(\cdot)$ on this set can be decomposed as

$$f(x) = \sum_{j=1}^{\infty} \langle \psi_j, f\rangle \psi_j(x); \qquad x \in X, \tag{3.36}$$

hence we can define the Nytström extension of $f(\cdot)$ to the rest of $\mathcal{M}$ to be

$$\overline{f}(\bar{x}) \triangleq \sum_{j=1}^{\delta} \langle \psi_j, f\rangle \overline{\psi}_j(\bar{x}); \qquad \bar{x} \in \overline{X}. \tag{3.37}$$

where $\overline{\psi}_j(\bar{x})$ is defined in (3.34) and δ is chosen considering stability issues. More specifically, choosing the parameter δ is application dependent which depends on both the manifold and the

function to be extended. In practical cases this parameter can be chosen by cross validation. For further discussion on this issue, interested readers are referred to [33]. It is worth mentioning that the vector field $\boldsymbol{r}$ which is computed on the set X can also be extended to $\mathcal{M}$ by treating each element of $\boldsymbol{r}$ as a function defined on the set X and using (3.37). The overall algorithm for embedding and function extension is summarized in Table 3.1, where $\boldsymbol{I}$ and $\mathbf{1}$ are respectively the identity matrix and a vector of ones with appropriate sizes, respectively.

3.3 Experimental Results

In this section, we examine the performance of the proposed algorithm in retrieving the geometry, the vector field, and the sampling density using toy examples. We also evaluate the performance of the proposed algorithm in extending an empirical function. Using directed graph embedding as a pre-process procedure, we also consider an application of the proposed directed graph embedding algorithm to a web page classification problem.

For the first simulation, let $\mathcal{M}$ be a one dimensional manifold embedded in R^3 which is given by the following parametric representation

$$\mathcal{M} = \left\{ x \in \mathrm{R}^3; x = g(\theta) = \begin{bmatrix} (2+\cos(8\theta))\cos(\theta) \\ (2+\cos(8\theta))\sin(\theta) \\ \sin(8\theta) \end{bmatrix}; \qquad 0 \le \theta \le 2\pi \right\}. \tag{3.38}$$

500 points are sampled from the manifold by sampling θ uniformly from the interval $[0\ \ 2\pi]$. This sampling procedure corresponds to sampling from the manifold by the following distribution [38]

$$p(x) = p(g(\theta)) = \alpha\sqrt{(\cos(8\theta)+2)^2+64}, \tag{3.39}$$

where α is a normalization factor such that $\int_{\mathcal{M}} p(x)dx = 1$. The similarity kernel is given by (3.1)-(3.3) where we choose $\varepsilon = 0.2$ and the vector field $\boldsymbol{r}(x,y)$ is given by

$$\boldsymbol{r}(x,y) = w(x) + w(y), \tag{3.40}$$

where

$$w(x) = w(g(\theta)) = 0.2 \begin{bmatrix} -8\sin(8\theta)\cos(\theta) - \sin(\theta)(\cos(8\theta)+2) \\ \cos(\theta)(\cos(8\theta)+2) - 8\sin(8\theta)\sin(\theta) \\ 8\cos(8\theta) \end{bmatrix}^T. \tag{3.41}$$

The results of Algorithm 3.2 and the method presented in [93] are depicted in Figure 3.1. It is obvious that the two algorithms are capable of retrieving correctly the geometry, the vector field and the sampling distribution. In this figure, the first and the second rows represent the results of the proposed algorithm and the method presented in [93], respectively. The left column shows the manifold together with the vector field. The embedding using the first two nondegenerate eigenvectors together with the estimated vector field are depicted in the second column. The actual and the estimated sampling distributions are represented in the third column. In order to show the effectiveness of the Algorithm 3.1 in extending the embedding and the vector field to new data points, we generate 1000 random points uniformly from the interval $[0\ 2\pi]$ and obtain the corresponding points on the manifold using (3.38). Then we compute their distances from the 500 points used in the first part of this experiment. The embedding is then extended using Algorithm 3.1. The result is depicted in Figure 3.2. The results of embedding extension using Nystöm extension and geometric harmonics for extending the vector field are also shown in Figure 3.2. Using the definition of the vector field in this experiment (i.e. equations (3.40)-(3.41)) it can be easily verified that the estimated vector field must be tangent to the embedding. Figure 3.2 (a) shows that the estimated vector field using the proposed algorithm is nearly tangent to the embedding while the estimated vector field using geometric harmonics is not. This is due to the essential assumption of symmetry of the kernel in geometric harmonics which does not held in this experiment. As it can be seen from the results, there exists a clear error between actual and estimated sampling distribution for both methods. Although this error is smaller in the proposed method it is still significant. In order to reduce this error, while the number of available data is constant, one must increase the kernel width (i.e. increase the parameter ε). While increasing the kernel width decreases the error in estimating the sampling distribution, it increases the error in embedding. In this simulation we have chosen the kernel width such that the embedding is close to the actual embedding which causes error in estimating the sampling distribution.

In the second experiment we show the advantage of the proposed method in function extension. Let $\mathcal{M}$ be the two dimensional interval $\mathcal{M} = \{x = [x_1, x_2]; -1 \leq x_1, x_2 \leq 1|\}$, and the

asymmetric similarity kernel be defined by

$$k_\varepsilon(x,y) = \begin{cases} \exp(-\frac{\|x-y\|^2}{\varepsilon^2}) & x_1 \leq y_1 \\ 0 & \text{otherwise} \end{cases}, \tag{3.42}$$

where $\varepsilon = 0.5$. Let X be a set of 100 points extracted uniformly from $\mathcal{M}$. We evaluate the function $f(x) = 1 + \sin(x_1)\cos(x_2)$ on this set. The test set $\overline{X}$ is built by deriving 4000 samples from the manifold uniformly. We use both the proposed method and the geometric harmonics for extending the function on these points. In the extension phase, we assume that we only know the distance of the points in $\overline{X}$ to X. The other distances (i.e. the distance of the points in X to $\overline{X}$) are assumed to be unknown. The results of function extension are depicted in Figure 3.3. It is obvious that the proposed method performs much better than the geometric harmonics. We attribute that to the tacit assumption of kernel symmetry in the construction of the geometric harmonics, which is not the case in this simulation. The accuracy of the function extension was quantified by the normalized root mean square error (NRMSE), given by

$$\text{NRMSE} = \sqrt{\frac{\sum_{\overline{x}\in\overline{X}}(f(\overline{x}) - \overline{f}(\overline{x}))^2}{\sum_{\overline{x}\in\overline{X}} f^2(\overline{x})}}. \tag{3.43}$$

The NRMSE for the aforementioned function extension wad 0.0717, compared to 0.2137 for the geometric harmonics approach.

In order to show the advantage of the proposed directed graph embedding, we consider the web-page classification problem. The World Wide Web can be regarded as a weighted directed graph with binary weights. In this graph each vertex (i.e. v_i) corresponds to a web-page (i.e. web-page number i) and directed edge from v_i to v_j has unity weight if there exist a link from page i to page j and zero otherwise. Using our notation in the previous section a corpus containing n web pages can be modelled with a weighted graph whose affinity matrix $K = \{K_{i,j}\}_{i,j=1}^n$ is given by

$$K_{i,j} = \begin{cases} 1 & i = j \text{ or there exist a link from page } i \text{ to page } j \\ 0 & \text{otherwise} \end{cases}. \tag{3.44}$$

In this simulation, we used the WebKb data set. This data set contains a subset of the WWW-pages collected from computer science departments of various universities in January 1997 by the World Wide Knowledge Base (Web-Kb) project of the Carnegie Mellon University (CMU) text learning group. The 1051 pages contains pages from the four universities Cornell

(243), Texas (254), Washington (298) and Wisconsin (256). We use three different directed graph embedding as a pre-process procedure, and then use K-means clustering to classify to four different categories. The affinity matrix and the results are depicted in Figures 3.4 (a) and (b), respectively. We compare our classification results to those of the methods due to Joncas-Meila (presented in [93]) and Chen-Yang-Tang (presented in [22]). The mean success rates versus number of training samples are depicted in Figure 3.4 (b). The mean success rate is computed as follows. Let m be the number of training samples. We randomly choose m web-pages from the corpus for training and use the rest as a test set. The success rate will be the ratio of the number of successfully classified samples to the size of the test set. For each m, we perform this train-test procedure 100 times and report the average. It is obvious the proposed method outperforms the competing methods.

In our last simulation, we use the proposed function extension method for supervised system identification using speech signals. Our goal is to identify a single pole stable discrete time filter (SPF) using arbitrary speech signals. More specifically, the problem is to identify the filter (i.e. finding the location of the pole of the filter) given a filtered version of speech signal. We assume that we only have N_t filtered speech signals $y_i(t) = x_i(t) * h_i(t); 1 \leq i \leq N_t$, where $x_i(t)$'s are arbitrary unknown speech signals, $h_i(t); 1 \leq i \leq N_t$ are the impulse responses of SPF's corresponding to poles $p_i; 1 \leq i \leq N_t$ and $*$ corresponds to discrete time convolution. The problem is to find $h_{test}(t)$ (or equivalently p_{test} the pole of the an unknown test filter) given $y_{test}(t) = x_{test}(t) * h_{test}(t)$ where x_{test} is also an unknown speech signal. Without loss of generality, we assume that all output of filters have the same length. Concisely, our function is a mapping from $y_i(t)$ to p_i which is to be extended to $y_{test}(t)$.

We define our asymmetric kernel as follows:

$$K(y_i, y_j) = K(p_i, p_j) = \begin{cases} \exp(-\frac{z}{\varepsilon}) & z > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.45}$$

$$z = \text{Kur}(h_j^{-1}(t) * y_i(t)) - \text{Kur}(x_i(t)) \tag{3.46}$$

where h_j^{-1} is the inverse of the impulse responses of SPF corresponding to poles p_j and $\text{Kur}(f(t))$ is the empirical kurtosis of the sequence $f(t)$.

In this simulation, we assume that the $0.2 \leq p_i \leq 0.6$. We uniformly extract 60 samples

from this interval and produce 60 SPF's. In order to generate training signals, for each SPF, we chose randomly a speech signal from the TIMIT [50] database of length 3 sec and filter it with the SPF. The similarity matrix is depicted in Figure 3.5 (a).

In the testing phase, we uniformly extract 100 samples from the interval $[0.2 \leq p_i^{test} \leq 0.6]$, and produce 100 test SPF's $h_i^{test}(t)$. In order to generate test signals, for each SPF, we chose randomly a speech signal $x_i^{test}(t)$ from the TIMIT [50] database of length 3 sec and not used in training step filter it with the SPF (i.e. $y_i^{test}(t) = h_i^{test}(t) * x_i^{test}(t)$). Then we use our algorithm and the algorithm presented in [93] for estimating the pole. The result is depicted in Figure 3.5 (b). The normalized root mean square error obtained of the algorithm presented in [93] and the proposed method are 14.31% and 2.05%, respectively. It is obvious the proposed method outperforms the competing method.

3.4 Conclusions

We have proposed a novel method based on Laplace operator for embedding a directed graph. We modelled the observed graph as a sample from a manifold endowed with a vector field, and designed an algorithm that separates and recovers the features of this process: the geometry of the manifold, the data density and the vector field. More specifically, we introduced an anisotropic diffusion type operator and we showed that under certain conditions this operator converges to a Laplace-Beltrami operator on the manifold. Using the fact that eigenfunctions of Laplace-Beltrami operator can be regarded as coordinates on the data set, we derived our embedding procedure. We also proposed a Nystöm type embedding extension and a novel technique for extending an empirical function on a directed graph. Simulation results demonstrated the advantage of the proposed method in function extension on a directed graph.

3.5 Appendix A: Proof of Lemma 3.2.1

In this appendix we provide the proof to Lemma 3.2.1. This lemma was introduced in [93] and the proof is given in a technical report. For consistency and since we used this lemma to prove Lemma 3.2.2, we provide the proof of this lemma here. We adopt the same notation as in the

proof of Lemma 8 in [32]. Let

$$G_\varepsilon f(x) = \int_{\mathcal{M}} k_\varepsilon(x,y) f(y) dy, \tag{3.47}$$

and $0 < \gamma < 1/2$. Then we have

$$\begin{aligned}
\left| \int_{y\in\mathcal{M}:\|x-y\|>\varepsilon^\gamma} k_\varepsilon(x,y) f(y) dy \right| &= \left| \int_{y\in\mathcal{M}:\|x-y\|>\varepsilon^\gamma} \left(1 + \frac{\boldsymbol{r}(x,y)}{2}\cdot(y-x)\right) h_\varepsilon(x,y) f(y) dy \right| \\
&\le \|f\|_\infty \left(1 + \left\|\frac{\boldsymbol{r}(x,y)}{2}\right\|_\infty\right) \left| \frac{1}{\varepsilon^{\frac{d}{2}}} \int_{y\in\mathcal{M}:\|x-y\|>\varepsilon^\gamma} \|x-y\| h\left(\frac{\|x-y\|^2}{\varepsilon}\right) dy \right| \\
&= \|f\|_\infty \left(1 + \left\|\frac{\boldsymbol{r}(x,y)}{2}\right\|_\infty\right) \left| \frac{1}{\varepsilon^{\frac{d}{2}}} \int_{y\in\mathcal{M}:\|y\|>\varepsilon^\gamma} \|y\| h\left(\frac{\|y\|^2}{\varepsilon}\right) dy \right| \\
&= \|f\|_\infty \left(1 + \left\|\frac{\boldsymbol{r}(x,y)}{2}\right\|_\infty\right) \left| \sqrt{\varepsilon} \int_{y\in\mathcal{M}:\|y\|>\varepsilon^{\gamma-1/2}} \|y\| h\left(\|y\|^2\right) dy \right| \\
&= C\|f\|_\infty \left(1 + \left\|\frac{\boldsymbol{r}(x,y)}{2}\right\|_\infty\right) \left| \sqrt{\varepsilon} \int_{y\in\mathrm{R}:y>\varepsilon^{\gamma-1/2}} y^d h\left(y^2\right) dy \right|,
\end{aligned} \tag{3.48}$$

where C is a constant and we used generalized spherical coordinates for obtaining the last equation. Using the exponential decay property of h, this term is exponentially small and bounded by $O(\varepsilon^{\frac{3}{2}})$. Hence, we have

$$G_\varepsilon f(x) = \int_{y\in\mathcal{M}} k_\varepsilon(x,y) f(y) dy = \int_{y\in\mathcal{M}:\|x-y\|\le\varepsilon^\gamma} \left(1 + \frac{\boldsymbol{r}(x,y)}{2}\cdot(y-x)\right) h_\varepsilon(x,y) f(y) dy + O(\varepsilon^{\frac{3}{2}}) \tag{3.49}$$

Now we can use a Taylor expansion since everything is localized around x. The two terms in the above integration can be expanded as follows. Let

$$g(y) = \left(1 + \frac{\boldsymbol{r}(x,y)}{2}\cdot(y-x)\right) f(y) \tag{3.50}$$

then

$$\begin{aligned}
g(y) &= g(x) + \sum_{i=1}^d s_i \frac{\partial g}{\partial s_i}(x) + \frac{1}{2}\sum_{i=1}^d\sum_{j=1}^d s_i s_j \frac{\partial^2 g}{\partial s_i \partial s_j}(x) + Q_3(s_1, s_2, \cdots, s_d) + O(\varepsilon^2) \\
&= f(x) + \sum_{i=1}^d s_i \left[\frac{\partial}{\partial s_i}\left(1 + \frac{\boldsymbol{r}(x,y)}{2}\cdot(y-x)\right)\Big|_{y=x} f(x) + \frac{\partial f}{\partial s_i}(x) \left(1 + \frac{\boldsymbol{r}(x,y)}{2}\cdot(y-x)\right)\Big|_{y=x} \right] \\
&\quad + \frac{1}{2}\sum_{i=1}^d\sum_{j=1}^d s_i s_j \left[\frac{\partial^2}{\partial s_i \partial s_j}\left(1 + \frac{\boldsymbol{r}(x,y)}{2}\cdot(y-x)\right)\Big|_{y=x} f(x) + \frac{\partial f}{\partial s_j}(x) \frac{\partial}{\partial s_i}\left(1 + \frac{\boldsymbol{r}(x,y)}{2}\cdot(y-x)\right)\Big|_{y=x} \right. \\
&\qquad \left. + \frac{\partial^2 f}{\partial s_i \partial s_j}(x) \left(1 + \frac{\boldsymbol{r}(x,y)}{2}\cdot(y-x)\right)\Big|_{y=x} + \frac{\partial f}{\partial s_i}(x) \frac{\partial}{\partial s_j}\left(1 + \frac{\boldsymbol{r}(x,y)}{2}\cdot(y-x)\right)\Big|_{y=x} \right] \\
&\qquad + Q_3(u) + O(\varepsilon^2),
\end{aligned} \tag{3.51}$$

where s_i is the i-th geodesic coordinate and $Q_m(s_1, s_2, \cdots, s_d)$ denotes a homogenous polynomial of degree m of the variables $s_1, s_2, \cdots, s_d$ which depends on x and might change from line

to line. This expansion can be written in the local coordinates of the tangent plane at point x as

$$\begin{aligned} g(u) = f(x) + \sum_{i=1}^{d} u_i \left[\frac{\partial}{\partial s_i} \left(1 + \frac{\boldsymbol{r}(x,y)}{2} \cdot (y-x) \right) \Big|_{y=x} f(x) + \frac{\partial f}{\partial s_i}(x) \right] \\ + \frac{1}{2} \sum_{i=1}^{d} \sum_{j=1}^{d} u_i u_j \left[\frac{\partial^2}{\partial s_i \partial s_j} \left(1 + \frac{\boldsymbol{r}(x,y)}{2} \cdot (y-x) \right) \Big|_{y=x} f(x) + \frac{\partial f}{\partial s_j}(x) \frac{\partial}{\partial s_i} \left(1 + \frac{\boldsymbol{r}(x,y)}{2} \cdot (y-x) \right) \Big|_{y=x} \right. \\ \left. + \frac{\partial^2 f}{\partial s_i \partial s_j}(x) + \frac{\partial f}{\partial s_i}(x) \frac{\partial}{\partial s_j} \left(1 + \frac{\boldsymbol{r}(x,y)}{2} \cdot (y-x) \right) \Big|_{y=x} \right] \\ + Q_3(u) + O(\varepsilon^2), \end{aligned} \tag{3.52}$$

where u_i is the i-th local coordinate of the tangent plane at point x and u is the orthogonal projection of y on the tangent plane at point x. Similarly, using Lemma 7 of [32], the kernel can be Taylor expanded as

$$h\left(\frac{\|x-y\|^2}{\varepsilon} \right) = h\left(\frac{\|u\|^2}{\varepsilon} \right) + \left(\frac{Q_5(u) + Q_4(u)}{\varepsilon} \right) h'\left(\frac{\|u\|^2}{\varepsilon} \right) + O(\varepsilon^2). \tag{3.53}$$

Substituting (3.52) and (3.53) into (3.49), and changing the variable $(y-x) \to u$, we get

$$\begin{aligned} G_\varepsilon f(x) &= \int_{y \in \mathcal{M}: \|x-y\| \le \varepsilon^\gamma} \left(1 + \frac{\boldsymbol{r}(x,y)}{2} \cdot (y-x) \right) h_\varepsilon(x,y) f(y) dy + O(\varepsilon^{\frac{3}{2}}) \\ &= \frac{1}{\varepsilon^{\frac{d}{2}}} \int_{\|u\| \le \varepsilon^\gamma} g(u) h\left(\frac{\|u\|^2}{\varepsilon} \right) (1 + Q_2(u) + Q_3(u))\, du + O(\varepsilon^{\frac{3}{2}}). \end{aligned} \tag{3.54}$$

Using the exponential decay property of the kernel, we can extend the domain of integration to R^d. Hence, we have

$$\begin{aligned} G_\varepsilon f(x) &= \frac{1}{\varepsilon^{\frac{d}{2}}} \int_{u \in \mathrm{R}^d} g(u) h\left(\frac{\|u\|^2}{\varepsilon} \right) (1 + Q_2(u) + Q_3(u))\, du + O(\varepsilon^2) \\ &= \frac{1}{\varepsilon^{\frac{d}{2}}} \int_{u \in \mathrm{R}^d} \left\{ f(x) + \sum_{i=1}^{d} u_i \left[\frac{\partial}{\partial s_i} \left(1 + \frac{\boldsymbol{r}(x,y)}{2} \cdot (y-x) \right) \Big|_{y=x} f(x) + \frac{\partial f}{\partial s_i}(x) \right] \right. \\ &+ \frac{1}{2} \sum_{i=1}^{d} \sum_{j=1}^{d} u_i u_j \left[\frac{\partial^2}{\partial s_i \partial s_j} \left(1 + \frac{\boldsymbol{r}(x,y)}{2} \cdot (y-x) \right) \Big|_{y=x} f(x) + \frac{\partial f}{\partial s_j}(x) \frac{\partial}{\partial s_i} \left(1 + \frac{\boldsymbol{r}(x,y)}{2} \cdot (y-x) \right) \Big|_{y=x} \right. \\ &\left. \left. + \frac{\partial^2 f}{\partial s_i \partial s_j}(x) + \frac{\partial f}{\partial s_i}(x) \frac{\partial}{\partial s_j} \left(1 + \frac{\boldsymbol{r}(x,y)}{2} \cdot (y-x) \right) \Big|_{y=x} \right] \right\} \times \\ &\left[h\left(\frac{\|u\|^2}{\varepsilon} \right) + \frac{(Q_5(u) + Q_4(u))}{\varepsilon} h'\left(\frac{\|u\|^2}{\varepsilon} \right) \right] \times \\ &[1 + Q_2(u) + Q_3(u)]\, du + O(\varepsilon^2). \end{aligned} \tag{3.55}$$

Setting the integration of the odd function to zero, we get

$$
\begin{aligned}
G_\varepsilon f(x) = \frac{1}{\varepsilon^{\frac{d}{2}}} \int_{u\in \mathrm{R}^d} \Bigg\{ & f(x) + \frac{1}{2}\sum_{i=1}^{d} u_i^2 \left[\frac{\partial^2}{\partial s_i^2}\left(1+\frac{\boldsymbol{r}(x,y)}{2}\cdot(y-x)\right)\Bigg|_{y=x} f(x) \right. \\
& \left. +2\frac{\partial f}{\partial s_i}(x)\,\frac{\partial}{\partial s_i}\left(1+\frac{\boldsymbol{r}(x,y)}{2}\cdot(y-x)\right)\Bigg|_{y=x} + \frac{\partial^2 f}{\partial s_i^2}(x)\right]\Bigg\} \times \\
& \left[h\left(\frac{\|u\|^2}{\varepsilon}\right) + \frac{Q_4(u)}{\varepsilon} h'\left(\frac{\|u\|^2}{\varepsilon}\right)\right] \times \\
& [1+Q_2(u)]\,du + O(\varepsilon^2). \qquad (3.56)
\end{aligned}
$$

Changing variables according to $u \to \sqrt{\varepsilon}u$ we have

$$
\begin{aligned}
G_\varepsilon f(x) = \int_{u\in \mathrm{R}^d} \Bigg\{ & f(x) + \frac{1}{2}\sum_{i=1}^{d} \varepsilon u_i^2 \left[\frac{\partial^2}{\partial s_i^2}\left(1+\frac{\boldsymbol{r}(x,y)}{2}\cdot(y-x)\right)\Bigg|_{y=x} f(x) \right. \\
& \left. +2\frac{\partial f}{\partial s_i}(x)\,\frac{\partial}{\partial s_i}\left(1+\frac{\boldsymbol{r}(x,y)}{2}\cdot(y-x)\right)\Bigg|_{y=x} + \frac{\partial^2 f}{\partial s_i^2}(x)\right]\Bigg\} \times \\
& \left[h\left(\|u\|^2\right) + \varepsilon Q_4(u) h'\left(\|u\|^2\right)\right] \times \\
& [1+\varepsilon Q_2(u)]\,du + O(\varepsilon^2) \\
= f(x) & \int_{u\in \mathrm{R}^d} h\left(\|u\|^2\right) du + \varepsilon f(x) \int_{u\in \mathrm{R}^d} \left(Q_2(u) h\left(\|u\|^2\right) + Q_4(u) h'\left(\|u\|^2\right)\right) du + \\
& \frac{\varepsilon}{2}\sum_{i=1}^{d} \left[\frac{\partial^2}{\partial s_i^2}\left(1+\frac{\boldsymbol{r}(x,y)}{2}\cdot(y-x)\right)\Bigg|_{y=x} f(x) + 2\frac{\partial f}{\partial s_i}(x)\,\frac{\partial}{\partial s_i}\left(1+\frac{\boldsymbol{r}(x,y)}{2}\cdot(y-x)\right)\Bigg|_{y=x} + \frac{\partial^2 f}{\partial s_i^2}(x)\right] \times \\
& \int_{u\in \mathrm{R}^d} u_1^2 h\left(\|u\|^2\right) dt + O(\varepsilon^2). \qquad (3.57)
\end{aligned}
$$

It is easy to verify that

$$
\nabla\cdot\boldsymbol{r} = \sum_{i=1}^{d} \frac{\partial^2}{\partial s_i^2}\left(1+\frac{\boldsymbol{r}(x,y)}{2}\cdot(y-x)\right)\Bigg|_{y=x} \qquad (3.58)
$$

$$
\nabla f\cdot\boldsymbol{r} = \sum_{i=1}^{d} 2\frac{\partial f}{\partial s_i}(x)\,\frac{\partial}{\partial s_i}\left(1+\frac{\boldsymbol{r}(x,y)}{2}\cdot(y-x)\right)\Bigg|_{y=x} \qquad (3.59)
$$

$$
\triangle f = \sum_{i=1}^{d} \frac{\partial^2 f}{\partial s_i^2}(x), \qquad (3.60)
$$

where $\nabla\cdot\boldsymbol{r}$, ∇f and $\triangle f$ are the divergence of the vector field $\boldsymbol{r}$, the gradient of function $f(\cdot)$ and the Laplace-Beltrami operator on the manifold, respectively. Substituting (3.58)-(3.60) into

(3.57) we have

$$\begin{aligned} G_\varepsilon f(x) &= f(x)\int_{u\in\mathrm{R}^d} h\left(\|u\|^2\right)du + \varepsilon f(x)\int_{u\in\mathrm{R}^d}\left(Q_2(u)h\left(\|u\|^2\right)+Q_4(u)h'\left(\|u\|^2\right)\right)du + \\ &\quad \frac{\varepsilon}{2}\left(\nabla\cdot\boldsymbol{r}+\nabla f\cdot\boldsymbol{r}+\triangle f\right)\int_{u\in\mathrm{R}^d} u_1^2 h\left(\|u\|^2\right)dt + O(\varepsilon^2) \\ &= m_0 f(x) + \varepsilon\frac{m_2}{2}\left(\omega(x)f(x)+\triangle f(x)+\boldsymbol{r}\cdot\nabla f(x)+f(x)\nabla\cdot\boldsymbol{r}\right)+O(\varepsilon^2). \end{aligned} \tag{3.61}$$

where

$$m_0 = \int_{\mathrm{R}^d} h(\|u\|^2)du \tag{3.62}$$

$$m_2 = \int_{\mathrm{R}^d} u_i^2 h(\|u\|^2)du \tag{3.63}$$

$$\omega(x) = \frac{2}{m_2}\int_{u\in\mathrm{R}^d}\left(Q_2(u)h\left(\|u\|^2\right)+Q_4(u)h'\left(\|u\|^2\right)\right)du. \tag{3.64}$$

In order to prove (3.18), note that

$$\begin{aligned} \int_{\mathcal{M}} k_\varepsilon(y,x)f(y)dy &= \int_{\mathcal{M}}\left(1+\frac{\boldsymbol{r}(y,x)}{2}\cdot(x-y)\right)h_\varepsilon(y,x)f(y)dy \\ &= \int_{\mathcal{M}}\left(1-\frac{\boldsymbol{r}(y,x)}{2}\cdot(y-x)\right)h_\varepsilon(y,x)f(y)dy, \end{aligned} \tag{3.65}$$

Since $\boldsymbol{r}(y,x)$ and $h_\varepsilon(y,x)$ are assumed to be symmetric, we have

$$\begin{aligned} \int_{\mathcal{M}} k_\varepsilon(y,x)f(y)dy &= \int_{\mathcal{M}}\left(1-\frac{\boldsymbol{r}(y,x)}{2}\cdot(y-x)\right)h_\varepsilon(y,x)f(y)dy \\ &= \int_{\mathcal{M}}\left(1-\frac{\boldsymbol{r}(x,y)}{2}\cdot(y-x)\right)h_\varepsilon(x,y)f(y)dy \\ &= m_0 f(x) + \varepsilon\frac{m_2}{2}\left(\omega(x)f(x)+\triangle f(x)-\boldsymbol{r}\cdot\nabla f(x)-f(x)\nabla\cdot\boldsymbol{r}\right)+O(\varepsilon^2). \end{aligned} \tag{3.66}$$

where the last equation is simply obtained by substituting $-\boldsymbol{r}(y,x)$ instead of $\boldsymbol{r}(y,x)$ in (3.17).

3.6 Appendix B: Proof of Theorem 3.2.3

In this appendix we provide the proof to Theorem 3.2.3. First we prove (3.28). Note that

$$\begin{aligned} T^{aa}_{\varepsilon,\alpha} f(x) &= \int_{\mathcal{M}} t_{k,\varepsilon,\alpha}(x,y) f(y) p(y) dy \\ &= \int_{\mathcal{M}} \frac{k^{(\alpha)}_{\varepsilon}(x,y)}{d_{k,\varepsilon}(x)} f(y) p(y) dy \\ &= \frac{\int_{\mathcal{M}} k^{(\alpha)}_{\varepsilon}(x,y) f(y) p(y) dy}{\int_{\mathcal{M}} k^{(\alpha)}_{\varepsilon}(x,y) p(y) dy}, \end{aligned} \tag{3.67}$$

where

$$k^{(\alpha)}_{\varepsilon}(x,y) = \frac{k_{\varepsilon}(x,y)}{p^{\alpha}_{k,\varepsilon}(x) p^{\alpha}_{k,\varepsilon}(y)}. \tag{3.68}$$

Using Lemma 3.2.1 and the definition of $p_{k,\varepsilon}(x)$ in (3.7) we have

$$\begin{aligned} p_{k,\varepsilon}(x) &= \int_{\mathcal{M}} k_{\varepsilon}(x,y) p(y) dy \\ &= p(x) + \varepsilon \left(\omega(x) p(x) + \triangle p(x) + \boldsymbol{r} \cdot \nabla p(x) + p(x) \nabla \cdot \boldsymbol{r} \right) \end{aligned} \tag{3.69}$$

up to a term of order $O(\varepsilon^2)$. Hence,

$$\frac{1}{p^{\alpha}_{k,\varepsilon}(x)} = p^{-\alpha}_{k,\varepsilon}(x) = p^{-\alpha}(x) \left(1 - \alpha\varepsilon \left(\omega(x) + \frac{\triangle p(x)}{p(x)} + \frac{\boldsymbol{r} \cdot \nabla p(x)}{p(x)} + \nabla \cdot \boldsymbol{r} \right) \right) \tag{3.70}$$

up to a term of order $O(\varepsilon^2)$. We define

$$K^{\alpha}_{\varepsilon} \phi(x) = \int_{\mathcal{M}} k^{(\alpha)}_{\varepsilon}(x,y) \phi(y) p(y) dy. \tag{3.71}$$

Then using the definition of $k^{\alpha}_{\varepsilon}(x,y)$, we have

$$\begin{aligned} K^{\alpha}_{\varepsilon} \phi(x) &= \int_{\mathcal{M}} \frac{k_{\varepsilon}(x,y)}{p^{\alpha}_{k,\varepsilon}(x) p^{\alpha}_{k,\varepsilon}(y)} \phi(y) p(y) dy \\ &= p^{-\alpha}_{k,\varepsilon}(x) \int_{\mathcal{M}} k_{\varepsilon}(x,y) \phi(y) \frac{p(y)}{p^{\alpha}_{k,\varepsilon}(y)} dy. \end{aligned} \tag{3.72}$$

Substituting the approximation of $p^{-\alpha}_{k,\varepsilon}(x)$ obtained in (3.70) into (3.72) we have

$$K^{\alpha}_{\varepsilon} \phi(x) = p^{-\alpha}_{k,\varepsilon}(x) \int_{\mathcal{M}} k_{\varepsilon}(x,y) \phi(y) p^{1-\alpha}(y) \left(1 - \alpha\varepsilon \left(\omega(y) + \frac{\triangle p(y)}{p(y)} + \frac{\boldsymbol{r} \cdot \nabla p(y)}{p(y)} + \nabla \cdot \boldsymbol{r} \right) \right) dy. \tag{3.73}$$

Defining

$$g(x) = \omega(x) + \frac{\triangle p(x)}{p(x)} + \frac{\boldsymbol{r}\cdot\nabla p(x)}{p(x)} + \nabla\cdot\boldsymbol{r}, \tag{3.74}$$

and using Lemma 3.2.1 we have

$$\begin{aligned}
K_\varepsilon^\alpha\phi(x) &= p_{k,\varepsilon}^{-\alpha}(x)\int_{\mathcal{M}} k_\varepsilon(x,y)\phi(y)\phi(y)p^{1-\alpha}(y)\,(1-\alpha\varepsilon g(y))\,dy \\
&= p_{k,\varepsilon}^{-\alpha}(x)\left[\phi(x)p^{1-\alpha}(x)\,(1-\alpha\varepsilon g(x)) + \varepsilon\left(\omega(x)\left(\phi(x)p^{1-\alpha}(x)\,(1-\alpha\varepsilon g(x))\right)\right.\right. \\
&\qquad + \triangle\left(\phi(x)p^{1-\alpha}(x)\,(1-\alpha\varepsilon g(x))\right) \\
&\qquad \left.\left. +\boldsymbol{r}\cdot\nabla\left(\phi(x)p^{1-\alpha}(x)\,(1-\alpha\varepsilon g(x))\right) + \left(\phi(x)p^{1-\alpha}(x)\,(1-\alpha\varepsilon g(x))\right)\nabla\cdot\boldsymbol{r}\right)\right].
\end{aligned} \tag{3.75}$$

Discarding terms of order $O(\varepsilon^2)$ we have

$$K_\varepsilon^\alpha\phi = p_{k,\varepsilon}^{-\alpha}\left[\phi p^{1-\alpha}(1-\alpha\varepsilon g) + \varepsilon\left(\omega\phi p^{1-\alpha} + \triangle\left(\phi p^{1-\alpha}\right) + \boldsymbol{r}\cdot\nabla\left(\phi p^{1-\alpha}\right) + \phi p^{1-\alpha}\nabla\cdot\boldsymbol{r}\right)\right], \tag{3.76}$$

where we have dropped the argument of each function (i.e (x)) for simplicity. Substituting $g(x)$ from (3.74) into (3.76) we have

$$\begin{aligned}
K_\varepsilon^\alpha\phi &= p_{k,\varepsilon}^{-\alpha}\left[\phi p^{1-\alpha} - \varepsilon\left(\alpha\phi p^{1-\alpha}\omega + \alpha\phi p^{1-\alpha}\frac{\triangle p}{p} + \alpha\phi p^{1-\alpha}\frac{\boldsymbol{r}\cdot\nabla p}{p} + \alpha\phi p^{1-\alpha}\nabla\cdot\boldsymbol{r}\right)\right. \\
&\qquad \left. +\varepsilon\left(\omega\phi p^{1-\alpha} + \triangle\left(\phi p^{1-\alpha}\right) + \boldsymbol{r}\cdot\nabla\left(\phi p^{1-\alpha}\right) + \phi p^{1-\alpha}\nabla\cdot\boldsymbol{r}\right)\right] \\
&= p_{k,\varepsilon}^{-\alpha}\left[\phi p^{1-\alpha} + \varepsilon\left((1-\alpha)p^{1-\alpha}(\phi\omega + \phi\nabla\cdot\boldsymbol{r}) - \alpha\phi\left(\frac{\triangle p}{p^\alpha} + \frac{\boldsymbol{r}\cdot\nabla p}{p^\alpha}\right) + \triangle\left(\phi p^{1-\alpha}\right) + \boldsymbol{r}\cdot\nabla\left(\phi p^{1-\alpha}\right)\right)\right] \\
&= p_{k,\varepsilon}^{-\alpha}p^{1-\alpha}\left[\phi + \varepsilon\left((1-\alpha)(\phi\omega + \phi\nabla\cdot\boldsymbol{r}) - \alpha\phi\left(\frac{\triangle p}{p} + \frac{\boldsymbol{r}\cdot\nabla p}{p}\right) + \frac{\triangle\left(\phi p^{1-\alpha}\right)}{p^{1-\alpha}} + \frac{\boldsymbol{r}\cdot\nabla\left(\phi p^{1-\alpha}\right)}{p^{1-\alpha}}\right)\right].
\end{aligned} \tag{3.77}$$

Using equation (3.67), the definition of the operator K_ε^α in (3.71) and equation (3.77), we get

$$\begin{aligned}
T_{\varepsilon,\alpha}^{aa}\phi &= \frac{K_\varepsilon^\alpha\phi(x)}{K_\varepsilon^\alpha 1} \\
&= \frac{\phi + \varepsilon\left((1-\alpha)(\phi\omega + \phi\nabla\cdot\boldsymbol{r}) - \alpha\phi\left(\frac{\triangle p}{p} + \frac{\boldsymbol{r}\cdot\nabla p}{p}\right) + \frac{\triangle\left(\phi p^{1-\alpha}\right)}{p^{1-\alpha}} + \frac{\boldsymbol{r}\cdot\nabla\left(\phi p^{1-\alpha}\right)}{p^{1-\alpha}}\right)}{1 + \varepsilon\left((1-\alpha)(\omega + \nabla\cdot\boldsymbol{r}) - \alpha\left(\frac{\triangle p}{p} + \frac{\boldsymbol{r}\cdot\nabla p}{p}\right) + \frac{\triangle\left(p^{1-\alpha}\right)}{p^{1-\alpha}} + \frac{\boldsymbol{r}\cdot\nabla\left(p^{1-\alpha}\right)}{p^{1-\alpha}}\right)}.
\end{aligned} \tag{3.78}$$

Utilizing the approximation $(1+\varepsilon a)^{-1} \approx 1-\varepsilon a$ and discarding terms of order $O(\varepsilon^2)$ we get

$$\begin{aligned}
T^{aa}_{\varepsilon,\alpha}\phi &= \phi + \varepsilon\left((1-\alpha)(\phi\omega+\phi\nabla\cdot\boldsymbol{r}) - \alpha\phi\left(\frac{\triangle p}{p}+\frac{\boldsymbol{r}\cdot\nabla p}{p}\right) + \frac{\triangle\left(\phi p^{1-\alpha}\right)}{p^{1-\alpha}} + \frac{\boldsymbol{r}\cdot\nabla\left(\phi p^{1-\alpha}\right)}{p^{1-\alpha}}\right) \\
&\quad - \phi\varepsilon\left((1-\alpha)(\omega+\nabla\cdot\boldsymbol{r}) - \alpha\left(\frac{\triangle p}{p}+\frac{\boldsymbol{r}\cdot\nabla p}{p}\right) + \frac{\triangle\left(p^{1-\alpha}\right)}{p^{1-\alpha}} + \frac{\boldsymbol{r}\cdot\nabla\left(p^{1-\alpha}\right)}{p^{1-\alpha}}\right) \\
&= \phi + \varepsilon\left(\frac{\triangle\left(\phi p^{1-\alpha}\right)}{p^{1-\alpha}} - \frac{\triangle\left(p^{1-\alpha}\right)}{p^{1-\alpha}}\phi + \frac{\boldsymbol{r}\cdot\nabla\left(\phi p^{1-\alpha}\right)}{p^{1-\alpha}} - \frac{\boldsymbol{r}\cdot\nabla\left(p^{1-\alpha}\right)}{p^{1-\alpha}}\phi\right) \\
&= \phi + \varepsilon\left(\triangle\phi + 2\frac{\nabla\phi\cdot\nabla p^{1-\alpha}}{p^{1-\alpha}} + \boldsymbol{r}\cdot\nabla\phi\right), \qquad (3.79)
\end{aligned}$$

where for obtaining the last equation we have used the following identities

$$\triangle(fg) = g\triangle f + f\triangle g + 2\nabla f\cdot\nabla g \qquad (3.80)$$

$$\nabla(fg) = f\nabla g + g\nabla f. \qquad (3.81)$$

Using the assumption $p = e^{-U}$, the definition of $L^{aa}_{\varepsilon,\alpha}$ in (3.26) and taking the limit when $\varepsilon \to 0$, we get

$$\begin{aligned}
\lim_{\varepsilon\to 0} L^{aa}_{\varepsilon,\alpha} f &= \lim_{\varepsilon\to 0}\frac{1-T^{aa}_{\varepsilon,\alpha}f}{\varepsilon} \\
&= \triangle f + 2\frac{\nabla f\cdot\nabla e^{-(1-\alpha)U}}{e^{-(1-\alpha)U}} + \boldsymbol{r}\cdot\nabla f \\
&= \triangle f - 2(1-\alpha)\nabla f\cdot\nabla U + \boldsymbol{r}\cdot\nabla f, \qquad (3.82)
\end{aligned}$$

where we have used $\nabla e^{-(1-\alpha)U} = -(1-\alpha)e^{-(1-\alpha)U}\nabla U$.

To prove (3.29) note that, similarly to the proof of the first part, all the results are based on a Taylor approximation of the kernel used in the integration operator used for obtaining the diffusion operator $T^{aa}_{\varepsilon,\alpha}$. In order to obtain the limit of infinitesimal generator of the Markov chain (i.e. $L^{ss}_{\varepsilon,\alpha}$) constructed from $T^{ss}_{\varepsilon,\alpha}$, note that the Taylor approximation of the kernel used in the integration operator for obtaining the diffusion operator $T^{ss}_{\varepsilon,\alpha}$ is the same as that of $T^{aa}_{\varepsilon,\alpha}$ when $\boldsymbol{r} = 0$ and ε is replaced by 2ε. Hence, using exactly the same procedure that led to (3.82), the result of (3.29) can be obtained.

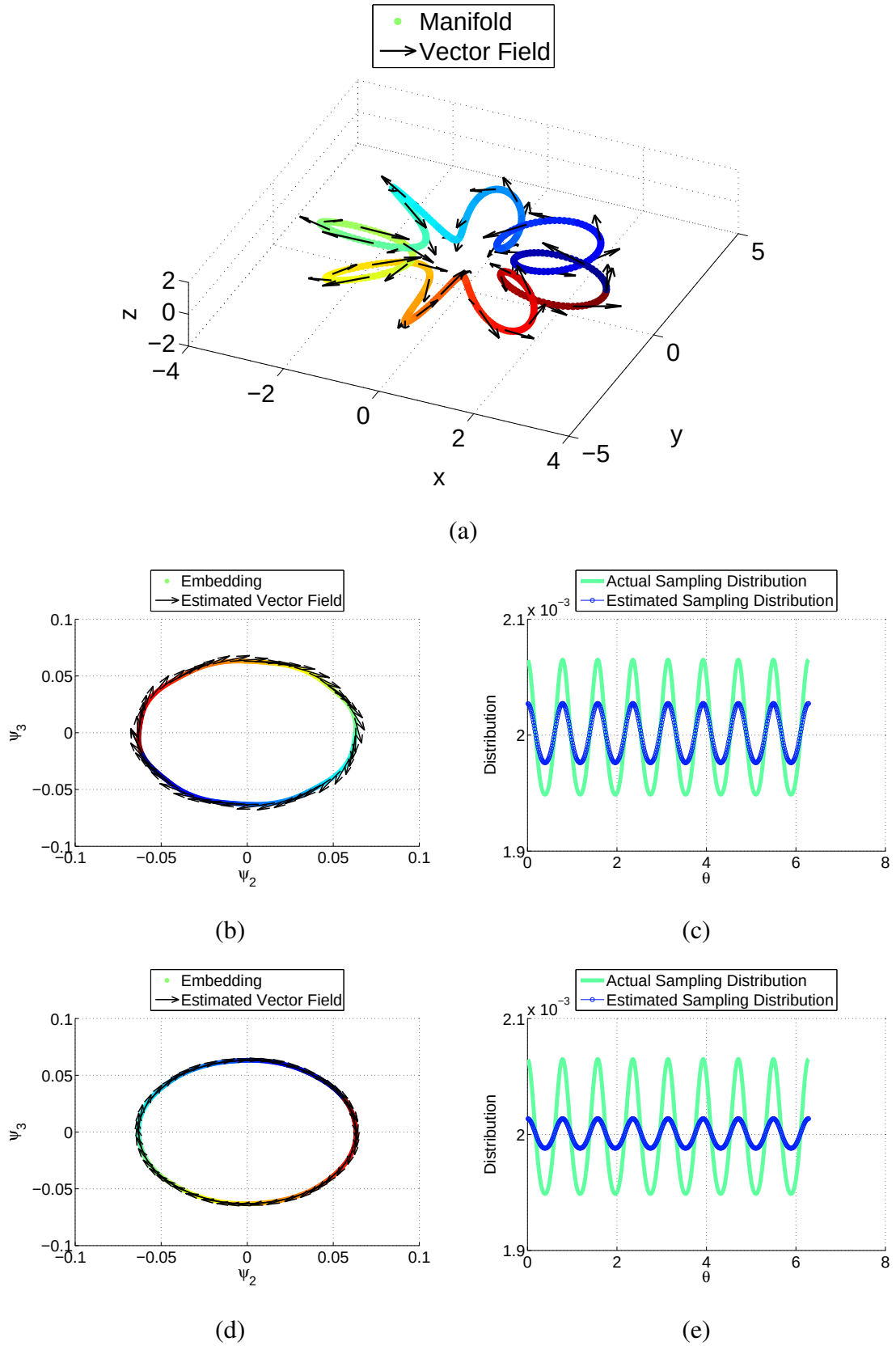
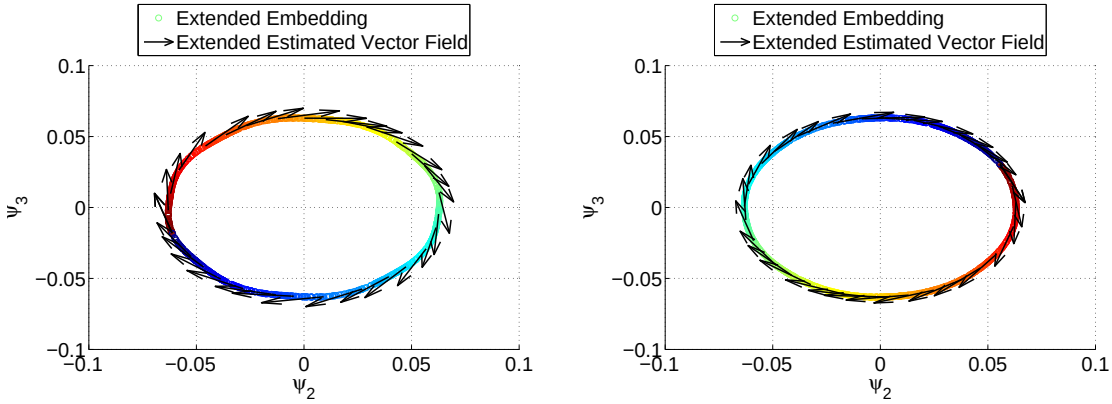
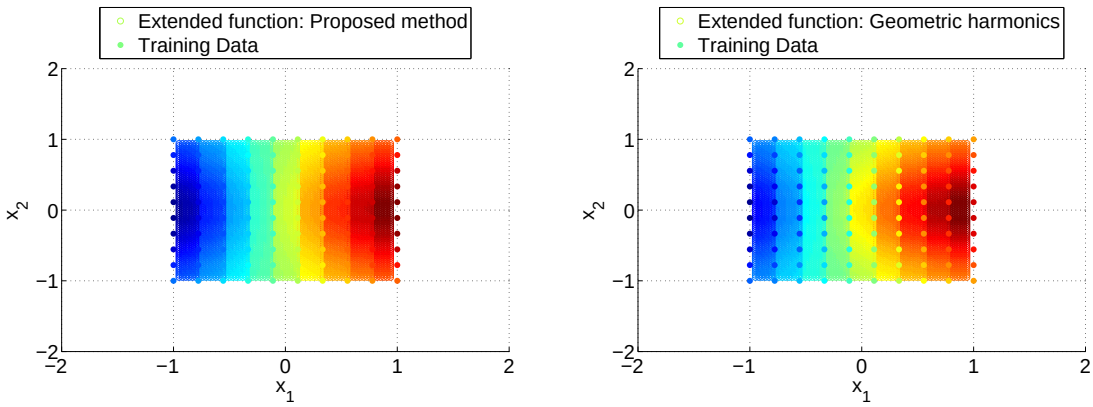


Figure 3.1: Comparison of the Algorithm 3.2 (second row) and the method proposed in [93] (third row): (a) One dimensional manifold embedded in R^3, (b)-(c) The embedding using the first two nondegenerate eigenvectors together and the estimated vector field, respectively. (d)-(e) The embedding using the first two nondegenerate eigenvectors together and the estimated vector field, respectively.



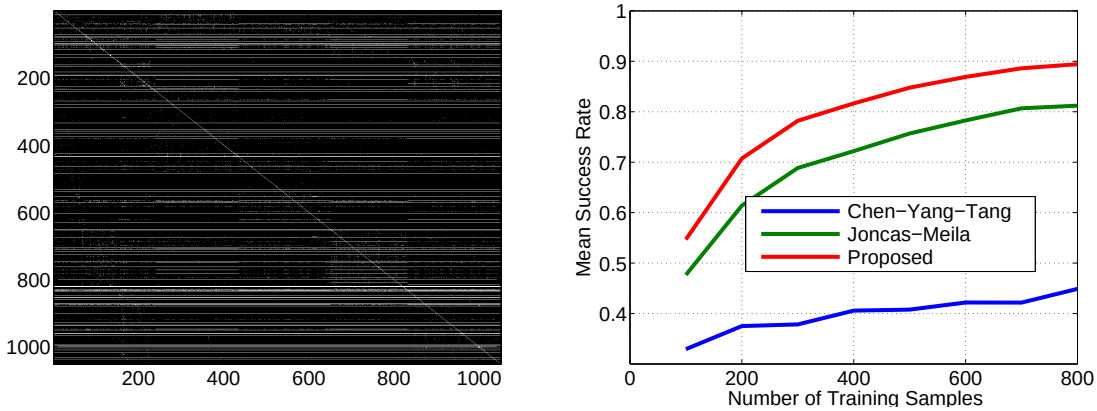
(a) (b)

Figure 3.2: Comparison of the Algorithm 3.1 (a) and the method proposed in [27] (b) for embedding and vector field extension



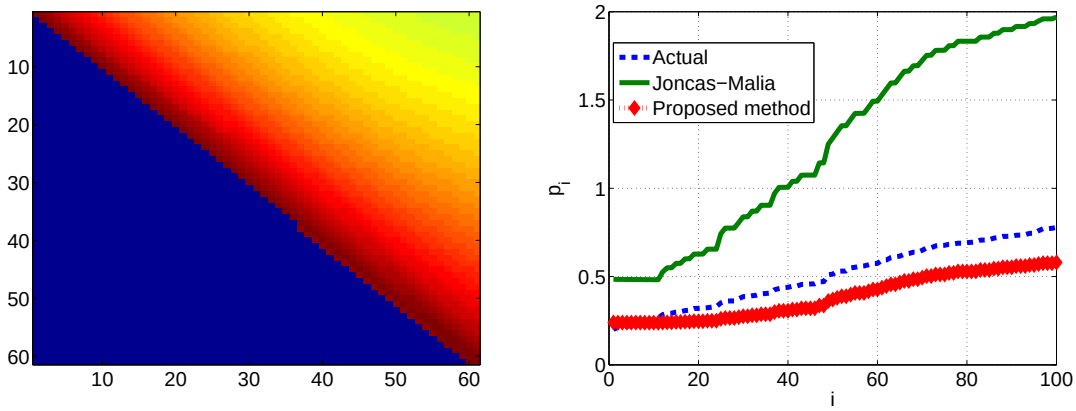
(a) (b)

Figure 3.3: Comparison of the Algorithm 3.1 (a) and the geometric harmonics (b) for function extension.



(a) (b)

Figure 3.4: Affinity matrix (a) and the comparison of the proposed method and the the methods due to Joncas-Meila (presented in [93]) and Chen-Yang-Tang (presented in [22] for web page classification.



(a) (b)

Figure 3.5: Affinity matrix (a) and the comparison of the proposed method and the the method due to Joncas-Meila (presented in [93]) for supervised system identification.

Table 3.1: Proposed Algorithm for Embedding and Function Extension on a Directed Graph

Algorithm 2: Embedding and Function Extension

Input: Affinity matrix $K \in \mathrm{R}^{n\times n}$ where n is the number of points in the training set, $W \in \mathrm{R}^{n\times 1}$ the affinity between the test point and the training points, $\psi \in \mathrm{R}^{n\times\delta}$ the eigenvectors of the Laplacian on the training data (obtained by Algorithm 1), $\Lambda \in \mathrm{R}^{\delta\times\delta}$ a diagonal matrix of eigenvalues (obtained by Algorithm 1), $R \in \mathrm{R}^{n\times m}$ the vector field components on the training points in the direction of the corresponding coordinates of the embedding, and $f \in \mathrm{R}^{n\times 1}$ the function value on the training set.

1. $\overline{\psi}(\bar{x}) = WK(\mathrm{diag}(WK\mathbf{1}))^{-1}\psi(\boldsymbol{I}-\Lambda)^{-1}$
2. $\overline{f}(\bar{x}) = f\psi\overline{\psi}(\bar{x})^T$
3. $\overline{R}(\bar{x})_i = R_i\psi\overline{\psi}(\bar{x})^T$; where $i = 1,2,\cdots,m$ and $\overline{R}(\bar{x})_i$ is the ith element of the extension of the vector field to $\bar{x}$ and $R_i \in \mathrm{R}^{1\times n}$ is the ith row of matrix R.

Table 3.2: Proposed Algorithm for Finding the Embedding, the Sampling Density and the Vector Field Representing the Directed Graph

Algorithm 1: Directed Graph Embedding

Input: Affinity matrix $K \in \mathrm{R}^{n\times n}$ where n is the number of points and embedding dimension m ($m \leq d$).

1. $S = KK^T$
2. $P_s = \text{diag}\left(\sum_{i=1}^{n} S_{i,j}\right)$
3. $S^1 = P_s^{-1} S P_s^{-1}$
4. $D_s = \text{diag}\left(\sum_{i=1}^{n} S_{i,j}^1\right)$
5. $T_s = D_s^{-1} S^1$
6. $L_s = \frac{I - T_s}{2\varepsilon}$
7. Compute $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ the eigenvalues of the matrix L_s and $\psi \in \mathrm{R}^{n\times n}$ a matrix of orthonormal columns containing the right eigenvector of L_s. Eigenvectors 2 to $m+1$ of L_s (i.e. 2nd through $(m+1)$th column of ψ) are the m coordinates of the embedding.
8. Compute $\Lambda = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)$, a diagonal matrix of eigenvalues.
9. Compute π the left eigenvector of L_s with eigenvalue 0.
10. $\pi = \frac{\pi}{\sum_i \pi_i}$ is the density distribution over the embedding.
11. $P_a = \text{diag}\left(\sum_{i=1}^{n} K_{i,j}\right)$
10. $K^1 = P_a^{-1} K P_a^{-1}$
11. $D_a = \text{diag}\left(\sum_{i=1}^{n} K_{i,j}^1\right)$
12. $T_a = D_a^{-1} S^1$
13. $L_a = \frac{I - T_a}{\varepsilon}$
14. $R = (L_a - L_s)\,\psi$. Columns 2 to $m+1$ of R are the vector field components in the direction of the corresponding coordinates of the embedding.

Chapter 4

Out-of-Sample Extension of Band-Limited Functions on Homogeneous Manifolds Using Diffusion Maps

4.1 Introduction

Supervised learning is a machine learning task of inferring a function from labeled training data [81]. The type of data and properties of the function are application dependent. One of the most basic supervised learning problem in the field of signal processing is sampling and reconstruction of a band-limited function. When the data points are real numbers and the function is band-limited, the classical Nyquist theorem [106] states that, the function can be perfectly reconstructed from its values on equally spaced points of reals, if the sampling rate is sufficiently high. The values of the function on points other than sampling points can be exactly calculated using SINC interpolator. Schoenberg [102] used cardinal splines for the reconstruction formula. There, it is shown that a band-limited function can be reconstructed from its values sampled at high enough rate (Nyquist rate) as accurately as needed using cardinal splines of sufficiently high degree. This result was further generalized to the case of nonuniform sampling by Lyubarskii and Madych [78]. More specifically, they showed that a band-limited function $f(x)$, whose Fourier transform is compactly supported between $[-\pi,\pi]$ can be completely reconstructed using spline functions, from its samples $f(x_n)$ taken at sampling points x_n, in the case when the functions $\exp(jx_n\omega)$, form a Riesz basis for $L_2([-\pi,\pi])$.

Pesenson [94] generalized the concept of band-limited functions to the case that the domain of the function is a homogeneous manifold and introduced the spectral entire functions of exponential type and Lagrangian splines on homogeneous manifolds. He also showed that on manifolds, the reconstruction of irregularly sampled spectral entire functions of exponential type (from now on band-limited functions) by splines is possible, as long as the distance between points of a sampling sequence is small enough.

Recently, using a different point of view, Coifman and Lafon [33] proposed a simple scheme, based on the Nyström method for supervised learning and extending empirical functions defined on a set X to a larger set $\bar{X}$. The extension process involves the construction of a specific family of functions termed geometric harmonics. These functions constitute a generalization of the prolate spheroidal wave functions of Slepian in the sense that they are optimally concentrated on X. Although being a powerful tool for function extension, this scheme does not make use of unlabeled data to improve the approximation of the Laplace-Beltrami operator on the manifold.

Supervised learning can be also regarded as the problem of function extension. The central dogma for studying the problem of function extension on manifolds is that the distribution of natural data is non-uniform and concentrates around low-dimensional structures. The shape (geometry) of the distribution can be exploited for efficient learning. As a justification for manifold assumption of natural data, see Jansen and Niyogi [63] for speech signals and Donoho and Grimes [41] for images. Geometrically derived methods have been used in applications such as image clustering [46, 133], image completion [52], speech enhancement in presence of transient noise [120], voice activity detection in presence of transient noise [85], linear and nonlinear independent component analysis [109, 110], parametrization of linear systems [121], and single channel source localization [119].

Zhu et al. [138] introduced an approach for supervised learning which is based on a Gaussian random field model. Labeled and unlabeled data were represented as vertices in a weighted graph, with edge weights encoding the similarity between instances. The learning problem was then formulated in terms of a Gaussian random field on this graph, where the mean of the field was characterized in terms of harmonic functions, and was efficiently obtained using matrix methods or belief propagation. In [89], it is shown that this method becomes ill posed as the number of unlabeled points tends to infinity . This observation was the motivation for Zhou

et al. [136] to address the semi-supervised learning problem and propose a solution by using regularization based on an iterated Laplacian, which is equivalent to a higher order Sobolev semi-norm. Their proposed solution can alternatively be viewed as a generalization of the thin plate spline to an unknown sub-manifold in high dimensions.

In most practical applications, besides the function, the data manifold is also unknown and just some labeled data (the points that the value of function is known on them, i.e. sampling points) and unlabeled data (the points that the value of function on them must be determined, i.e. interpolation points) are available. This means that the sampling theorem on manifolds [94] cannot be utilized directly to learn and extend the function to unlabeled data because in [94], the manifold is assumed to be known.

In order to be able to use a sampling theorem on manifolds, one needs to completely know the manifold. This means that the Laplace-Beltrami operator on the manifold must be known and can be computed for every function. Many manifold learning algorithms have been introduced during the last decade, among them one can name isomap [123], Locally-linear embedding (LLE) [103], Laplacian eigenmaps [14] and diffusion maps [32]. Diffusion maps leverage the relationship between heat diffusion and a random walk on a graph. The heat diffusion on manifold, is the diffusion process whose infinitesimal generator is the Laplace-Beltrami operator. In [32], an analogy is drawn between the diffusion operator on a manifold and a Markov transition matrix operating on functions defined on the graph whose nodes were sampled from the manifold. It is also shown that one can approximate the Laplace-Beltrami operator using appropriately normalized Markov transition matrix.

In this chapter, we propose a novel technique for supervised learning when the data is assumed to be located on a manifold. More specifically, we use diffusion maps as a tool for manifold learning and approximating the Laplace-Beltrami operator on a manifold. Next, we use sampling theorem of band-limited functions on manifolds [94] to extend the function onto the interpolation points. The solution coincides with the method proposed in [136], hence gives another justification for the method presented in [136]. This chapter is organized as follows. In Section 4.2, we formulate the problem and introduce our function extension algorithm. In Section 4.3, we evaluate the performance of our method and compare it to several available function extension methods. We also discuss some applications of the proposed method. We conclude the chapter in Section 4.4.

4.2 Problem Formulation

Let $\mathcal{M}$ be a C^∞-homogeneous manifold and Δ be the Laplace-Beltrami operator in the corresponding Hilbert space $L^2(\mathcal{M})$. We say that a function $f(\cdot)$ from $L^2(\mathcal{M})$ is ω_0-band-limited if the function satisfies the Bernstein inequality [94]

$$\|\Delta^{k/2} f\| \leq \omega_0^k \|f\| \tag{4.1}$$

for every natural even k, where $\|f\|$ denotes $L^2(\mathcal{M})$ norm. Using Parseval's theorem, it can be easily verified that in the special case where $\mathcal{M} = \mathrm{R}$, this definition is equivalent to the definition of band-limited functions (i.e. the Fourier transform is compactly supported in $[-\omega_0, \omega_0]$).

A set of points $Z_\lambda = \{\boldsymbol{x}_\gamma\}$, is called a sampling sequence if

(a) $\inf_{\gamma \neq \mu} \operatorname{dist}(\boldsymbol{x}_\gamma, \boldsymbol{x}_\mu) > 0$,

(b) Balls $B(\boldsymbol{x}_\gamma, \lambda)$ form a cover of $\mathcal{M}$, and

(c) $\lambda < (c_0 \omega_0)^{-1}$,

where c_0 is a manifold-dependent constant. In the case $\mathcal{M} = \mathrm{R}$, the last condition becomes the Nyquist sampling condition if the sampling is uniform. It can be shown [94] that any ω_0-band-limited function can be exactly reconstructed from its samples as long as the value of the function is known on a sampling sequence.

In [94], it is shown that given an ω_0-band-limited function $f(\cdot)$ on a d-dimensional manifold $\mathcal{M}$, $\varepsilon > 0$ and a sampling sequence Z_λ, there exists a function $\hat{f}^k$ such that

$$\|f - \hat{f}^k\| < \varepsilon \qquad ; \qquad k = 2^l d, \quad l \in \mathrm{N} \tag{4.2}$$

for a sufficiently large l. The function $\hat{f}^k$ is the solution of the following optimization problem

$$\begin{aligned} &\hat{f}^k = \arg\min_u \|\Delta^{k/2} u\| \\ &\text{s.t.} \\ &\hat{f}^k(\boldsymbol{x}_\gamma) = f(\boldsymbol{x}_\gamma); \qquad \forall \boldsymbol{x}_\gamma \in Z_\lambda. \end{aligned} \tag{4.3}$$

In [94], it has been shown that the above optimization problem has a unique solution for each k, but the author did not supply any algorithm for solving it in the general case. The main contribution of this chapter is to present a method for solving the above optimization problem in the case where no information about the manifold is available (i.e. the Laplace-Beltrami

operator is unknown). We assume that we have a valid sampling sequence (i.e. a sequence satisfying the conditions (a) through (c) above) so that we can exact reconstruction is possible. We also assume that we have enough (theoretically infinite but at least exponential in the intrinsic dimension of the manifold) unlabelled data the approximation of the Laplace-Beltrami is sufficiently accurate. This issue will be discussed later in more details.

Diffusion maps [32] is a powerful tool for nonlinear dimensionality reduction. Besides being a useful means for dimensionality reduction, it can be used for manifold learning. Using anisotropic kernel for constructing diffusion maps, Nadler et al. [88], described a random walk construction that in the limit of infinite data recovers the Laplace-Beltrami (heat) operator on the manifold on which the data resides, regardless of the possibly non-uniform sampling of points on it. This normalization is therefore best suited for learning the geometry of the data-set, as it separates the geometry of the manifold from the statistics on it. In what follows we briefly overview the method and introduce a novel technique for solving the optimization problem in (4.3).

Let $X = \{\boldsymbol{x}_j\}_{j=1}^{J}$ be a finite data set sampled from a homogeneous manifold $\mathcal{M} \subset \mathrm{R}^P$ randomly sampled from some arbitrary probability distribution. Suppose that we are given a kernel $k : X \times X \to \mathrm{R}$ that is symmetric (i.e. $k(\boldsymbol{x},\boldsymbol{y}) = k(\boldsymbol{y},\boldsymbol{x})$) and is positive (i.e. $k(\boldsymbol{x},\boldsymbol{y}) \geq 0$). This kernel represents some notion of affinity or similarity between points of X. This kernel describes the relationship between pairs of points in the set X and in this sense, one can think of the data points as being the nodes of a symmetric graph whose weight function is specified by k. Without loss of generality, for a fixed value of ε (a meta-parameter of the algorithm), we define an isotropic diffusion kernel,

$$k_\varepsilon(\boldsymbol{x},\boldsymbol{y}) = \exp\left(\frac{-\|\boldsymbol{x}-\boldsymbol{y}\|^2}{\varepsilon}\right). \tag{4.4}$$

We construct an $J \times J$ similarity matrix $\boldsymbol{K}$ such that

$$\boldsymbol{K}_{i,j} = k_\varepsilon\left(\boldsymbol{x}_i,\boldsymbol{x}_j\right), \tag{4.5}$$

where $\boldsymbol{K}_{i,j}$ is the (i,j)-th element of matrix $\boldsymbol{K}$. Next, we normalize the similarity matrix as follows,

$$\boldsymbol{W} = \boldsymbol{D}^{-1}\boldsymbol{K}\boldsymbol{D}^{-1}, \tag{4.6}$$

where $\boldsymbol{D}$ is a diagonal matrix whose (i,i)-th element equals to the sum of the i-th row (or equivalently column) of $\boldsymbol{K}$ (i.e. $\boldsymbol{D} = \mathrm{diag}(\boldsymbol{K}\mathbf{1})$ where $\mathbf{1}$ is the column vector of ones). The

Laplacian matrix $\boldsymbol{L}$ is then obtained by

$$\boldsymbol{P} = \boldsymbol{D}_W^{-1}\boldsymbol{W}, \tag{4.7}$$

$$\boldsymbol{L} = \frac{\boldsymbol{I}-\boldsymbol{P}}{\varepsilon} \tag{4.8}$$

where $\boldsymbol{D}_W = \mathrm{diag}(\boldsymbol{W}\boldsymbol{1})$. It can be shown that, both $\boldsymbol{L}$ and $\boldsymbol{L}^T$ (where $(\cdot)^T$ denotes the transpose of a matrix or a vector) converge to the Laplace-Beltrami operator when the number of data, J, approaches infinity and ε approaches zero [32, 88]. Since our approximation of the Laplace-Beltrami operator needs to be symmetric, we choose

$$\boldsymbol{L}_s = \frac{\boldsymbol{L}+\boldsymbol{L}^T}{2}, \tag{4.9}$$

as a discrete approximation of the Laplace-Beltrami operator.

Let $X_s = \{\boldsymbol{x}_i^s\}_{i=1}^M$ be a set consisting of a labelled or training data (i.e. a sampling sequence) which satisfying the conditions (a) through (c), and let $\boldsymbol{f}^s$ be a column vector such that $\boldsymbol{f}_i^s = f(\boldsymbol{x}_i^s)$ where $\boldsymbol{f}_i^s$ is the i-th element of vector $\boldsymbol{f}^s$. Let $X_I = \{\boldsymbol{x}_i^I\}_{i=1}^N$ be a set consisting of unlabeled or testing data (i.e. an interpolation sequence). Let $\boldsymbol{f}^I$ be a column vector such that $\boldsymbol{f}_i^I = f(\boldsymbol{x}_i^I)$ where $\boldsymbol{f}_i^I$ is the i-th element of vector $\boldsymbol{f}^I$. The goal is to determine $\boldsymbol{f}^I$ given the value of a band-limited function $f(\cdot)$, on a sampling sequence (i.e. given vector $\boldsymbol{f}^s$). We assume $N+M$ is sufficiently large such that the approximation of the Laplace-Beltrami operator on the manifold is accurate enough (at least exponential in the intrinsic dimension of the manifold). Note that, here we have two independent sampling procedures. In order to learn the function on the manifold we need labeled data which means sampling the function defined on the manifold such that the conditions (a) through (c) are satisfied. In order to learn the manifold (i.e. approximating of the Laplace-Beltrami operator on the manifold), we need samples from the manifold, to learn the Laplace-Beltrami operator on the manifold. The diffusion maps use all the available data sampled from the manifold to find the approximation of Laplace-Beltrami operator, whether or not they are labeled.

Let $\boldsymbol{X}$ be a $(M+N)\times P$ matrix where each row represents a single data point. Without loss of generality we assume that the first M rows represent the sampling sequence (i.e. $\boldsymbol{X}_{i,:} = \boldsymbol{x}_i^s$ for $i = 1,2,\cdots,M$, where $\boldsymbol{X}_{i,:}$ is the i-th row of matrix $\boldsymbol{X}$) and the next N rows consist of interpolation points (i.e. $\boldsymbol{X}_{i+M,:} = \boldsymbol{x}_i^I$ for $i = 1,2,\cdots,N$). Let $\boldsymbol{f}$ be a column concatenation of $\boldsymbol{f}^s$ and $\boldsymbol{f}^I$ i.e.

$$\boldsymbol{f} = \begin{bmatrix} \boldsymbol{f}^s \\ \boldsymbol{f}^I \end{bmatrix} \tag{4.10}$$

Let K be the $(M+N) \times (M+N)$ similarity matrix computed by

$$\boldsymbol{K}_{i,j} = k_\varepsilon(\boldsymbol{X}_{i,:}, \boldsymbol{X}_{j,:}), \tag{4.11}$$

and let $\boldsymbol{L}_s$ be a discrete approximation of the Laplace-Beltrami operator computed using (4.6) through (4.9). Using $\boldsymbol{L}_s$, discrete approximation of the optimization problem in (4.3) is obtained as follows

$$\begin{aligned} &\hat{\boldsymbol{f}}^k = \arg\min_{\boldsymbol{f}} \|\boldsymbol{L}_s^{k/2}\boldsymbol{f}\| \\ &\text{s.t.} \\ &\hat{\boldsymbol{f}}_i^k = \boldsymbol{f}_i^s; \qquad I = 1, 2, \cdots, M, \end{aligned} \tag{4.12}$$

where $\hat{\boldsymbol{f}}_i^k$ and $\boldsymbol{f}_i^s$ are the i-th elements of $\hat{\boldsymbol{f}}^k$ and $\boldsymbol{f}^s$, respectively. The above optimization problem can be equivalently written as follows

$$\begin{aligned} &\hat{\boldsymbol{f}}^k = \arg\min_{\boldsymbol{f}} \boldsymbol{f}^T \boldsymbol{L}_s^k \boldsymbol{f} \\ &\text{s.t.} \\ &\boldsymbol{A}\hat{\boldsymbol{f}}^k = \boldsymbol{f}^s, \end{aligned} \tag{4.13}$$

where $\boldsymbol{A} = [\boldsymbol{I}_{M\times M} \vdots \boldsymbol{0}_{M\times N}]$ and $\boldsymbol{I}_{M\times M}$ and $\boldsymbol{0}_{M\times N}$ are the identity and zero matrices of appropriate sizes, respectively. This is a simple quadratic programming problem with equality constraints and can be solved easily using range space or null space approaches [76]. Using the null space method, the solution of the optimization problem in (4.13) is given by

$$\hat{\boldsymbol{f}}^k = \boldsymbol{A}\boldsymbol{f}_s - \boldsymbol{Z}\left(\boldsymbol{Z}^T\boldsymbol{L}_s^k\boldsymbol{Z}\right)^{-1}\left(\boldsymbol{Z}^T\boldsymbol{L}_s^k\boldsymbol{A}\boldsymbol{f}_s\right), \tag{4.14}$$

where $\boldsymbol{Z} \in \mathrm{R}^{(N+M)\times(N)}$ is the matrix whose columns span the null space of $\boldsymbol{A}$ and is given by

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{0}_{M\times N} \\ \boldsymbol{I}_{N\times N} \end{bmatrix}. \tag{4.15}$$

The overall function extension algorithm is summarized in Table 4.2.

The last important issues which must be addressed here are the determination of the power of Laplacian (i.e. k), the kernel width ε and the effect of number of samples on the performance of

the proposed methods. As is discussed in [94], if the Laplace-Beltrami operator is fully known, the error in function extension decreases by increasing k. Hence, in the proposed method, before k becomes too large, causing numerical issues, almost always larger k performs better. From the Sobolev embedding theorem, increasing k restricts solution space to be a smoother space, and from kernel point of view, increasing k corresponds to a better density adaptive kernel [136]. It can be shown that, in continuous case, the $L^2(\mathcal{M})$ norm of the error in function reconstruction decrease exponentially as k grows [94]. In practice, k can be chosen by cross-validation method.

The convergence to the continuous diffusion operator can be utilized for properly choosing the kernel width ε. In [58] and [30], it was proposed to automatically set the scale by examining a logarithmic scale of the sum of the kernel weights, without computing the spectral decomposition of the transition matrix. More specifically, using the fact that the logarithmic plot of $\sum_{i,j} k_\varepsilon(x_i, x_j)$ with respect to $\log \varepsilon$ cannot be linear for all ε's, they suggested to choose the kernel width ε from that linear region. See [30, 58] and [120] for further discussion.

The last issue to be discussed here is the effect of number of samples on the performance of the proposed methods. Several chapters provide rigorous estimates for the accuracy of the approximation of Laplace-Beltrami operator. For instance, it is shown in [32] that with high probability the error in approximation of Laplace-Beltrami operator using finite data is of the order $O\left(J^{-1/2}\varepsilon^{-d/4-1/2}\right)$ where J is the number of data points used for approximating the Laplace-Beltrami operator, d is the intrinsic dimension of the manifold and ε is the kernel width.

4.3 Applications, Simulation Results and Performance Evaluation

In this section, we evaluate the performance of the proposed method using simulations and compare our method to several available methods. In all simulations, the kernel width (i.e, ε) in all kernel based methods is chosen such that the normalized root mean square error (NRMSE) (to be defined shortly in (4.16)) is minimized. Although computation of NRMSE needs the knowledge of the actual function (which is the quantity we try to estimate), we use this method for all the methods to make our comparison fair. Choosing an appropriate kernel width is a

Table 4.1: Proposed Algorithm for Extension of Band-Limited Functions on Homogenous Manifolds Using Diffusion Maps

Input $X_s = \{\boldsymbol{x}_i^s\}_{i=1}^M$: The set containing a sampling sequence $\boldsymbol{f}^s$: A column vector such that $\boldsymbol{f}_i^s = f(\boldsymbol{x}_i^s)$. $X_I = \{\boldsymbol{x}_i^I\}_{i=1}^N$: The set containing the interpolation sequence. $k_\varepsilon(\cdot,\cdot)$: A kernel represents local similarity between points.
(1) Construct $X = [X_s; X_I]$ by column concatenation of sampling sequence and interpolation points. (2) Construct the similarity matrix $\boldsymbol{K}$ such that $\boldsymbol{K}_{i,j} = k_\varepsilon\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)$. (3) Let $\boldsymbol{W} = \boldsymbol{D}^{-1}\boldsymbol{K}\boldsymbol{D}^{-1}$, $\boldsymbol{P} = \boldsymbol{D}_W^{-1}\boldsymbol{W}$, $\boldsymbol{L} = \frac{\boldsymbol{I}-\boldsymbol{P}}{\varepsilon}$, $\boldsymbol{L}_s = \frac{\boldsymbol{L}+\boldsymbol{L}^T}{2}$, where $\boldsymbol{D} = \mathrm{diag}(\boldsymbol{K}\mathbf{1})$ and $\boldsymbol{D}_W = \mathrm{diag}(\boldsymbol{W}\mathbf{1})$. (4) Let $\hat{\boldsymbol{f}}^k = \boldsymbol{A}\boldsymbol{f}_s - \boldsymbol{Z}\left(\boldsymbol{Z}^T\boldsymbol{L}_s^k\boldsymbol{Z}\right)^{-1}\left(\boldsymbol{Z}^T\boldsymbol{L}_s^k\boldsymbol{A}\boldsymbol{f}_s\right)$, where $\hat{\boldsymbol{f}}^k$ is the estimate of the function on all data.
Output $\hat{\boldsymbol{f}}^I$: A column vector such that $\hat{\boldsymbol{f}}_i^I = \hat{f}(\boldsymbol{x}_i^I)$.

critical issue in most of kernel methods (like the proposed method or geometric harmonics), but dealing with this issue is beyond the scope of this chapter.

In the first example, we try to reconstruct a band-limited function defined on an interval of reals from uniformly sampled points. Note that an interval of reals is not a homogeneous manifold, but neglecting the effect of the end points it is a homogeneous manifold. We assume that the sampling rate is more than the Nyquist rate so exact reconstruction is possible. In this case, the optimal reconstruction scheme is a SINC interpolator. We try to reconstruct

$$f(x) = \sin(2\pi x) + \sin(4\pi x + \frac{2\pi}{5}) + \sin(6\pi x + \frac{\pi}{3}) + \cos(8\pi x)$$

from its samples. The sampling frequency was set to 10 Hz (a little bit more than the Nyquist rate which is 8 Hz). We try to reconstruct the function in interval $[0,10]$. We sample the manifold (i.e. the interval $[0,10]$) uniformly with frequency 100 Hz. The results of simulation are depicted in Figure 4.1. The normalized root mean square error (NRMSE) in extension is computed by

$$\text{NRMSE} = \sqrt{\frac{\sum_{\boldsymbol{x}\in X} |f(\boldsymbol{x}) - \hat{f}(\boldsymbol{x})|^2}{\sum_{\boldsymbol{x}\in X} |f(\boldsymbol{x})|^2}} \tag{4.16}$$

as a function of k, and is depicted in Figure 4.1. It is obvious that for a sufficiently large k, the MSE of the proposed method is lower than that of geometric harmonics. It is also apparent that the proposed method converges to the optimal solution (i.e. SINC interpolation) as k increases. Although theoretically increasing k improves the performance of the proposed function extension algorithm, it must be taken into account that increasing k while the number of available data (i.e. $N+M$) is constant, does not necessarily improve the overall performance (i.e. decreasing MSE in function extension; see the following simulations for an example). This is because of the fact that the estimation of Laplace-Beltrami operator near the boundaries of the manifold and regions with high curvature is not very accurate. Increasing k causes intense error in extension scheme because of error in estimating the Laplace-Beltrami operator near the boundaries and regions with high curvature.

In the second simulation, we try to reconstruct the same function as in the first simulation from nonuniform sampling points. In this case, a band-limited function defined on reals whose Fourier transform is compactly supported between $[-W,W]$ can be completely reconstructed using spline functions from its samples $f(x_n)$ taken at sampling points x_n, in the case when the functions $\exp(jx_n\omega)$ form a Riesz basis for $L_2([-W,W])$. Without loss of generality, suppose

that $\{x_n; n \in \mathrm{Z}\}$ is an increasing sequence. It can be shown that, the sequence $\exp(jx_n\omega)$ forms a Riesz basis for $L_2([-W,W])$, if $x_n = \frac{n\pi}{W} + r_n$ where $|r_n| < \frac{\pi}{4W}$ [64]. In this simulation, the sampling on the manifold is exactly the same as in the previous simulation (i.e. uniform samples on interval $[0,10]$ with frequency 100 Hz). Samples of the function are taken at the same points as in the previous simulation with random uniform displacement of maximum 0.025 (i.e. $x_n = \frac{n}{10} + r_n$ where r_n is uniformly distributed in $[-0.025, 0.025]$). The simulation results are depicted in Figure 4.2. NRMSE as a function of k is depicted in Figure 4.2. It is obvious that for k sufficiently large, the NRMSE achieved by the proposed method is lower than that obtained by geometric harmonics.

In the third simulation, we try to reconstruct the same function as in the second simulation from nonuniform sampling points while the domain is the unit circle in a two dimensional Euclidean space which is a homogeneous manifold. We choose 1500 points from the interval $[-\pi,\pi]$ sampled from normal distribution of zero mean and variance π. Then we embed them to the unit circle in a two dimensional Euclidean space. The function $f(x,y)$ is then defined as

$$f(x,y) = \sin(2\pi x) + \sin(4\pi x + \frac{2\pi}{5}) + \sin(6\pi x + \frac{\pi}{3}) + \cos(8\pi x). \tag{4.17}$$

The simulation results are depicted in Figure 4.3. NRMSE as a function of k is depicted in Figure 4.3. It is obvious that for k large enough, the NRMSE achieved by the proposed method is comparable to that obtained by geometric harmonics and much better than the method proposed in [138].

In the following example, we recover the parameters of an autoregressive-moving average (ARMA) system. Consider the following ARMA process of orders (p,q)

$$y(t) - \sum_{\ell=1}^{p} \boldsymbol{a}_\ell y(t-\ell) = x(t) + \sum_{i=1}^{q} \boldsymbol{b}_i x(t-i) \tag{4.18}$$

where $x(t)$ is a zero-mean white noise with variance σ_x^2 , and $\{\boldsymbol{a}_\ell\}$ and $\{\boldsymbol{b}_i\}$ are the AR and the MA coefficients, respectively. Such an ARMA process is commonly used in many signal processing applications. ARMA model is appropriate when a system is a function of a series of unobserved shocks (the MA part) as well as its own behaviour. For example, stock prices may be shocked by fundamental information as well as exhibiting technical trending and mean-reversion effects due to market participants.

An ARMA process can be viewed as a white noise filtered by a linear system, where the

corresponding transfer function is given by

$$H_{\boldsymbol{a},\boldsymbol{b}}(\omega) = \frac{1+\sum_{i=1}^{q} \boldsymbol{b}_i e^{-j\omega i}}{1-\sum_{\ell=1}^{p} \boldsymbol{a}_\ell e^{-j\omega \ell}}. \tag{4.19}$$

Let $S_{\boldsymbol{a},\boldsymbol{b}}(\omega)$ be the power spectral density (PSD) of the ARMA process given by

$$\begin{aligned} S_{\boldsymbol{a},\boldsymbol{b}}(\omega) &= \sigma_x^2 |H_{\boldsymbol{a},\boldsymbol{b}}(\omega)|^2 \\ &= \sigma_x^2 \frac{|1+\sum_{i=1}^{q} \boldsymbol{b}_i e^{-j\omega i}|^2}{|1-\sum_{\ell=1}^{p} \boldsymbol{a}_\ell e^{-j\omega \ell}|^2}. \end{aligned} \tag{4.20}$$

We observe in last equation, that the PSD depends only on the ARMA parameters (i.e. controlling parameters) when σ_x^2 assumed to be known. Consequently, the variations of the controlling parameters are conveyed by the PSD. Now, from (4.20), we can express the covariance function of the output signal as

$$c_{y_{\boldsymbol{a},\boldsymbol{b}}}(\tau) = \mathfrak{F}^{-1}\left\{S_{\boldsymbol{a},\boldsymbol{b}}(\omega)\right\}, \tag{4.21}$$

where $\mathfrak{F}^{-1}\{\cdot\}$ denotes the inverse Fourier transform. Hence, we can conclude that the PSD and consequently the covariance function of a specific output signal is a high dimensional data lying in a low dimensional manifold of dimension $p+q$ embedded in a high dimensional space.

Now if we look at each parameters of the model as function of the covariance function, our task is to extend this function from a given sampling sequence to an interpolating sequence. In this example, we examine the ability of the proposed algorithm to recover the parameters of ARMA$(1,1)$ processes. The parameters of the ARMA processes (i.e. $\boldsymbol{a}_1$ and $\boldsymbol{b}_1$) are uniformly sampled from the rectangle $[0, 0.5\ 0.3, 1]$ (i.e. $0 \leq \boldsymbol{a}_1 \leq 0.3$ and $0.5 \leq \boldsymbol{b}_1 \leq 1$). The process noise used in this simulation is chosen to be independent identically distribute white Gaussian noise with unity variance. We generate 1000 ARMA processes with different parameters and compute 20 correlation coefficients. From these 1000, 20-dimensional correlation coefficients, we uniformly (with respect to the sampling on parameter space) chose 16 of them for training sequence and utilize the proposed algorithm to extended the function on the rest of the points. We use a Gaussian kernel for similarity computation with scale parameter set to $\varepsilon = 0.5$, i.e.

$$k_\varepsilon\left(c_{y_{\boldsymbol{a}_1^i,\boldsymbol{b}_1^i}}(\tau), c_{y_{\boldsymbol{a}_1^j,\boldsymbol{b}_1^j}}(\tau)\right) = \exp\left(-\frac{\|c_{y_{\boldsymbol{a}_1^i,\boldsymbol{b}_1^i}}(\tau) - c_{y_{\boldsymbol{a}_1^j,\boldsymbol{b}_1^j}}(\tau)\|^2}{0.5}\right). \tag{4.22}$$

The results of the simulation are depicted in Figures 4.4 through 4.5. We compare our results to those of iterative search method widely used in estimating the parameters of the ARMA model

[115]. An iterative search algorithm minimizes a robustified quadratic prediction error criterion. The iterations are terminated either when the maximum number of iterations is reached, or when the expected improvement is less than the specified tolerance, or when a lower value of the criterion cannot be found. We have also compared our results with those of KNN-search and the method proposed in [138]. In Figures 4.4 and 4.5 the x-axis regards to the power of Laplacian for the proposed method or number of nearest neighbor in KNN-search method. As it can be seen from these Figures 4.4 and 4.5, the proposed method outperforms the competing methods for large enough k in estimating the parameters of AR part while have better or approximately the same performance as competing method in estimating the parameters of MA part.

In our last experiment, we examine the ability of the proposed method to recover the controlling parameters of acoustic channels, simulated using the image method. The propagation of a sound wave within an enclosure can be considered linear if the medium is homogeneous. Hence, the acoustic channel from a source to a microphone is obtained by solving the wave equation. However, this solution is difficult to be expressed analytically; hence, in practice usually some approximation is utilized. The image method, presented by Allen and Berkley [8] is one of the most common methods for this task. In this method, the acoustic channel between a source and a sensor in a rectangular room is approximated by a linear time invariant (LTI) finite impulse response (FIR) system.

In order to approximate the acoustic channels in typical rooms, Habets [56] has provided a software which simulates the acoustic channel between a source and a sensor using image method which has 12 controlling parameter. These controlling parameter consist of $\boldsymbol{\beta} = [\beta_{x_1}, \beta_{x_2}, \beta_{y_1} \beta_{y_2}, \beta_{z_1}, \beta_{z_2}]$ the reection coefcients of the six walls, $\boldsymbol{r}_s = [x_s, y_s, z_s]$ and $\boldsymbol{r}_m = [x_m, y_m, z_m]$ the source and the microphone location, respectively. Typical impulse responses consist of thousands of taps. In other words, each impulse response can be expressed as a vector in a high-dimensional space. However, as we have discussed earlier, the acoustic channel between a source and a microphone inside a rectangular room is controlled by a set of 12 controlling parameter. Hence, we can consider each impulse response as a point from a low dimensional manifold embedded in a high dimensional space. Now if we regard each controlling parameter as a function defined on the impulse response, and we know the value of these parameters on a sampling sequence, we can use the proposed method to extend these function on unlabeled data (i.e. the impulse responses whose controlling parameter is unknown) in order

to estimate these parameters.

In this experiment, we recover the reflection coefficients of two walls. We generate 20 training channels, where β_{x_1} is sampled uniformly from the interval [0.15 0.55] and we set $\beta_{x_2} = 0.7 - \beta_{x_1}$. The other four coefcients are set to 0.5. Then we simulate a room of size [4, 5, 3] meter. We place a microphone at $\boldsymbol{r}_m = [2,\ 1.5,\ 2]$, and a source at $\boldsymbol{r}_s = [2,\ 3.5,\ 2]$, distant 2 meters from the microphone. We then generate the acoustic impulse response of length 4096 samples and compute 30 correlation coefficients of each impulse response.

The test set is exactly obtained as in training set where for β_{x_1} we sampled 180 points uniformly from the interval [0.15 0.55] and we set $\beta_{x_2} = 0.7 - \beta_{x_1}$. All other parameters are set as in training procedure. We used a Gaussian kernel with kernel width $\varepsilon = 1.0710 \times 10^{-18}$.

The results of the simulation are depicted in Figures 4.6 through 4.9. We compare our results to those of KNN-search. In Figures 4.8 and 4.9 the x-axis regards to the power of Laplacian for the proposed method or number of nearest neighbor in KNN-search method. As it can be seen from these Figures 4.8 and 4.9, the proposed method outperforms the KNN-search method for large enough k in estimating the parameters of acoustic impulse response. It is worth mentioning that proposed method can be easily utilized to recover the direction of arrival of a random source in a room based on observations from a *single* microphone and training.

4.4 Conclusions

We have proposed a novel technique for supervised learning when the data lies on a manifold. We treated this problem as a problem of extending a band-limited function on a homogeneous manifold while the function and the manifold are both unknown. The only assumption for reconstructing the function is that we have enough labeled data to reconstruct the function (i.e. enough data points on the manifold on with known function values) and enough data to approximate the Laplace-Beltrami operator on the manifold. Diffusion maps were utilized in order to learn the manifold (i.e. approximating the Laplace-Beltrami operator on the manifold). Simulation results show the advantage of the proposed method over the widely-used geometric harmonics. The proposed method is insensitive to the sampling procedure on the manifold. More specifically, the proposed method is not sensitive to the distribution of points sampled on the manifold. This is related to the property that diffusion maps with appropriate normalization

recovered the Riemannian geometry of the data set.

An important issue which must be addressed here is the obtaining the constant c_0 (see condition (c) above). This constant is manifold-dependent and must be obtained appropriately to know if we have enough data for perfect reconstruction. The question of how to obtain this constant cannot be answered in general except, that for a compact manifold of dimension d the number of sampling points should be of order $\omega^{d/2}$, where ω is the bandwidth. See [94] for further discussion.

The last important issue which is interesting and needs more research is the behaviour of the proposed method in the case of $L^{\infty}(\mathcal{M})$ norm. Answering to this question is beyond the scope of this chapter and to the best of our knowledge it is an open problem. We conjecture that under this norm of the error is bounded. A special case of this problem has been answered in [105].

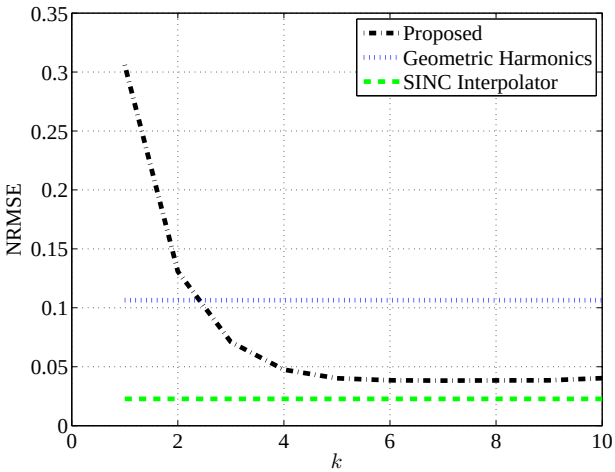


Figure 4.1: Normalized root mean square error (NRMSE) in function extension for different methods as a function of the parameter k.

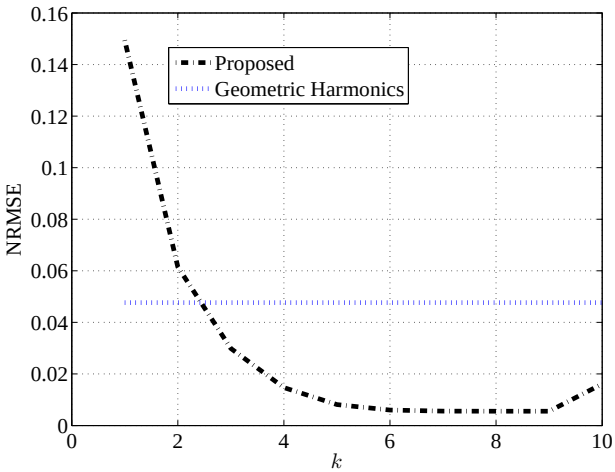


Figure 4.2: Normalized root mean square error (NRMSE) in function extension for different methods as a function of the parameter k.

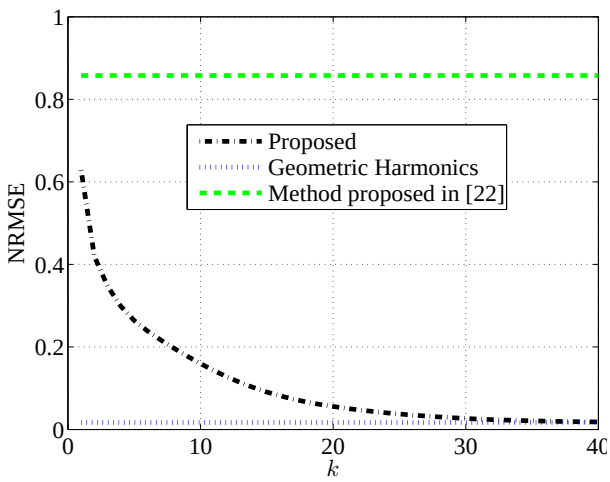


Figure 4.3: Normalized root mean square error (NRMSE) in function extension for different methods as a function of the parameter k.

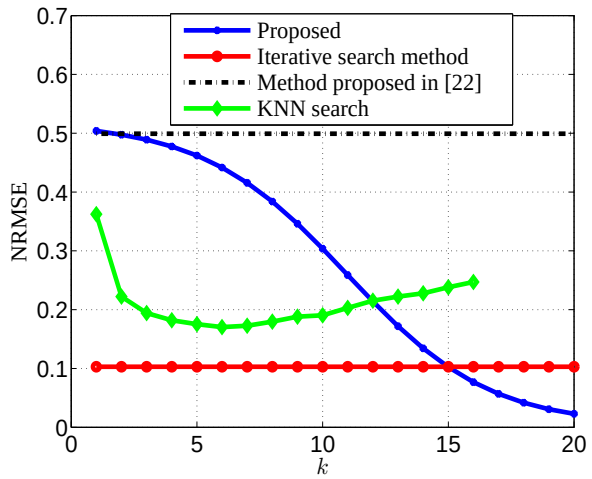


Figure 4.4: Normalized root mean square error (NRMSE) in estimating the parameters of the AR part for different methods as a function of the parameter k.

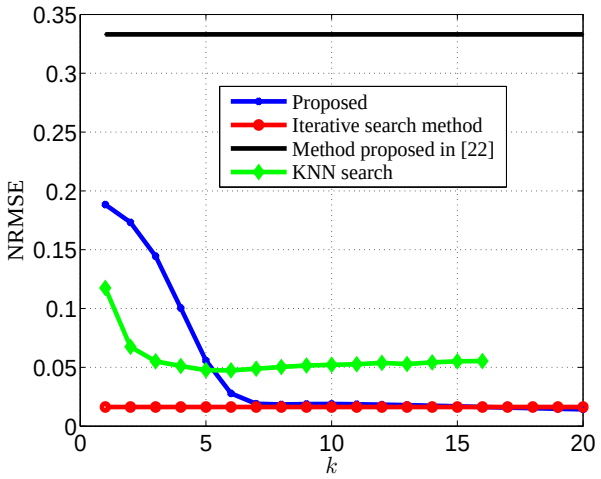


Figure 4.5: Normalized root mean square error (NRMSE) in estimating the parameters of the MA part for different methods as a function of the parameter k.

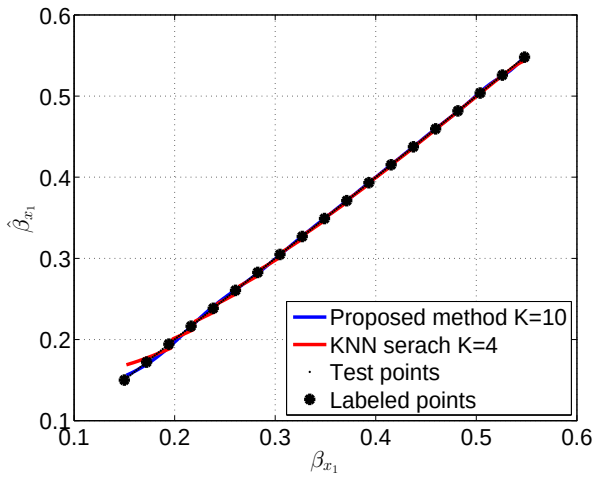


Figure 4.6: Estimated reflection coefficient β_{x_1} versus the true value for different methods.

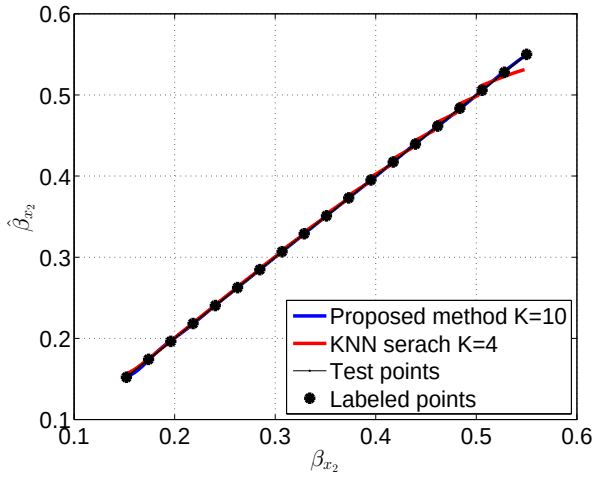


Figure 4.7: Estimated reflection coefficient β_{x_2} versus the true value for different methods.

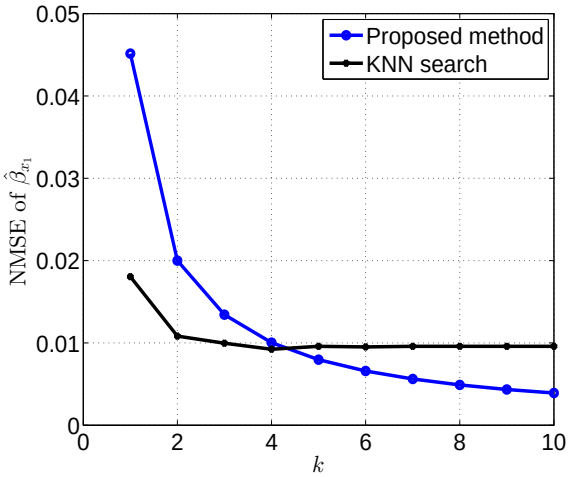


Figure 4.8: Normalized root mean square error (NRMSE) in estimating the reflection coefficient β_{x_1} for different methods as a function of the parameter k.

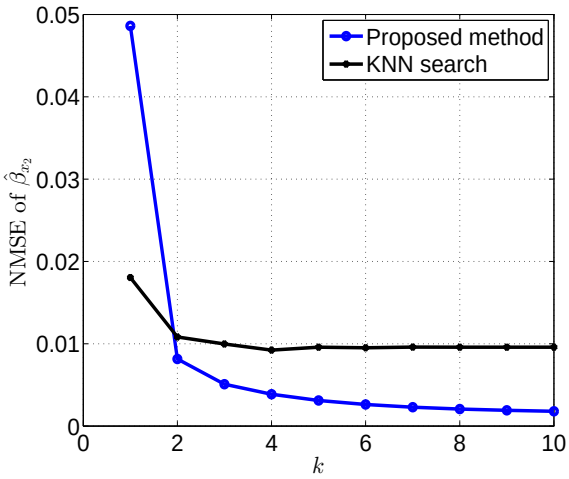


Figure 4.9: Normalized root mean square error (NRMSE) in estimating the reflection coefficient β_{x_2} for different methods as a function of the parameter k.

Chapter 5

Simultaneous Parameter Estimation and State Smoothing of Complex GARCH Process in the Presence of Additive Noise

5.1 Introduction

Parameter estimation is the backbone of all model-based signal processing algorithms. Having variability clustering and heavy tail property, Autoregressive Conditional Heteroscedasticity (ARCH) and Generalized ARCH (GARCH) models have recently been used in speech signal modeling. In speech enhancement in the Short Time Fourier Transform (STFT) domain, the most important quantity which is to be estimated correctly is the spectral variance, since having the true value of this quantity, one can use the minimum mean square error (MMSE) [43] or log spectral amplitude (LSA) [44] estimators, which are optimum in MMSE and LSA sense respectively, in order to estimate the speech spectral component. Cohen [25] modeled the speech signal in the STFT domain as a complex GARCH process and used this model for speech enhancement. He showed that the time varying variance of the speech signal can be estimated using a GARCH model with Gaussian innovations. Furthermore, Decision-Directed method of Ephraim and Malah [43] can be derived using GARCH modeling. In his work, the GARCH model is used to estimate the spectral variance of the speech signals. Tahmasbi et al. [118] used the GARCH model for the Voice Activity Detection (VAD). Abdolahi et al. [2] used the parameters of the GARCH model for speech recognition in Persian isolated digits. In all of

the above mentioned applications, it is assumed that the parameters of the GARCH model are known or can be estimated from a clean database.

Parameter estimation methods for ARCH and GARCH models, such as Quasi Maximum Likelihood (QML) [129], Two Stage Least Squares (TSLS) [20], and Constrained Two Stage Least Squares (CTSLS) [86], rely on the clean data. Since these models are mostly used in econometric where data can be assumed noise free, there was no effort to estimate the parameters of these models in presence of additive noise. In many speech signal processing algorithms, these methods are inefficient since the additive noise is inevitable. Hence, new methods for estimating ARCH and GARCH parameters in presence of noise are required. Poyiadjis et al. [95] introduced a method for ML parameter estimation of general state-space models using particle methods. The drawback of their method is its very high computational load, which makes this approach inapplicable to speech signal processing.

In this chapter, we propose a new ML based estimation method for estimating the parameters of GARCH model in presence of additive noise. We find the probability density function (pdf) of the observation data given the parameters and compute the maximum of this function. Specifically, we compute the gradient vector and the Hessian matrix of the probability density function of the data, and subsequently find the maximum of the probability density function numerically using the Newton method. The reminder of this chapter is organized as follows. In Section 5.2, we formulate the problem. In Section 5.3, we introduce our method and an adaptive version thereof. In Section 5.4, we apply our method for hidden state estimation. Finally in Section 5.5, we examine the performance of our method using simulations.

5.2 Problem formulation

A complex GARCH process of order (p,q) is defined as follows

$$\mathbf{x}_t = \sqrt{\boldsymbol{\lambda}_{t|t-1}} \odot \mathbf{v}_t \tag{5.1}$$

$$\boldsymbol{\lambda}_{t|t-1} = c_0\mathbf{1} + \sum_{i=1}^{p} a_i \mathbf{x}_{t-i} \odot \mathbf{x}_{t-i}^* + \sum_{j=1}^{q} b_j \boldsymbol{\lambda}_{t-j|t-j-1} \tag{5.2}$$

where $\mathbf{x}_t$ is a $k \times 1$ complex vector, $\mathbf{1}$ denotes a vector of ones, $\odot$ denotes a term-by-term multiplication, * denotes the conjugate transpose operator, k is the dimension of the model and $\mathbf{v}_t$'s are independent identically distributed complex Gaussian random vectors with zero mean

and identity covariance matrix, i.e.,

$$\mathbf{v}_t \sim \mathcal{CN}(0, \mathbf{I}) \,. \tag{5.3}$$

An ARCH model is a special case of the GARCH model with $b_j = 0 \forall j$.

It is worth mentioning that this model is a special case of a more complicated general model in which the c_0 is a $k \times 1$ vector and a_i's and b_j's are $k \times k$ matrices. Although it initially seems that the general model might be more successful in applications such as speech modeling (because it models the speech in each frequency bin using different GARCH models) one should take into account that the general model consists of $k^2 + k$ parameters with $k = 512$ or 1024 in speech processing applications. This large number of parameters makes the general model inapplicable in practice. Another issue which must be considered is that the process is really a vector process (not k separate scalar processes) because as we will see latter, all the vectors are combined with each other in order to jointly estimate the parameters using ML method.

Suppose that the observation signal is an ARCH or GARCH process corrupted with an additive zero-mean white complex Gaussian noise, i.e.,

$$\mathbf{y}_t = \mathbf{x}_t + \mathbf{n}_t \tag{5.4}$$

$$\mathbf{n}_t \sim \mathcal{CN}\left(0, \sigma^2 \mathbf{I}\right) \tag{5.5}$$

where $\mathbf{n}_t$ is the additive noise sequence. We assume that the noise variance σ^2 is unknown, and must be estimated from observations. Our objective is to find constrained estimates of the GARCH parameters and the noise variance (i.e., estimation of the vector of the parameters $\boldsymbol{\theta} = [c_0, a_1, ..., a_p, b_1, ..., b_q, \sigma^2]^T$, under the stationarity conditions of the model). The stationarity conditions for the complex GARCH model are given by [25]

$$c_0 > 0 \tag{5.6}$$

$$a_1, a_2, ..., a_p, b_1, b_2, ..., b_p \geq 0 \tag{5.7}$$

$$\sum_{i=1}^{p} a_i + \sum_{j=1}^{q} b_j < 1. \tag{5.8}$$

5.3 ML Estimation of the Parameters

In order to find the maximum likelihood estimate of the parameters, we must compute the logarithm of the likelihood function of the observations. Let $\mathbf{y}_0^\tau = \{\mathbf{y}_t | t = 0, 1, 2, ..., \tau\}$ be the set

of observations up to sample number τ. Using Bayes' formula, the logarithm of the likelihood function can be written as

$$\log\left[f(\mathbf{y}_0^{N-1};\boldsymbol{\theta})\right] = \sum_{t=1}^{N-1} \log\left[f\left(\mathbf{y}_t|\mathbf{y}_0^{t-1};\boldsymbol{\theta}\right)\right] + \log\left[f(\mathbf{y}_0;\boldsymbol{\theta})\right] \tag{5.9}$$

where N is the number of available observations. It should be noted that for a large sample data size the second term on the right side of (5.9) makes negligible contribution to the pdf and can be ignored. Let $\mathbf{x}_0^{\tau} = \{\mathbf{x}_t|t = 0,1,2,\ldots,\tau\}$ be the clean signal up to sample number τ. We denote the one-frame-ahead conditional variance of the clean signal as

$$\boldsymbol{\lambda}_{t|t-1} = E\left\{\mathbf{x}_t \odot \mathbf{x}_t^*|\mathbf{x}_0^{t-1};\boldsymbol{\theta}\right\}. \tag{5.10}$$

It is worth mentioning that in [25], it is shown that $\boldsymbol{\lambda}_{t|t-1}$ can be used as an estimate of the speech spectral variance. As in [25] we assume that, the past estimated conditional variances are sufficient statistics for the conditional variance estimation i.e.

$$E\left\{\mathbf{x}_t \odot \mathbf{x}_t^*|\mathbf{y}_0^t;\boldsymbol{\theta}\right\} = E\left\{\mathbf{x}_t \odot \mathbf{x}_t^*|\hat{\boldsymbol{\lambda}}_{t|t-1},\mathbf{y}_t;\boldsymbol{\theta}\right\}. \tag{5.11}$$

Here, we assume that the order of the model is $(1,1)$. We choose this order because this order is mostly used in the speech processing applications. Note that the one-frame-ahead conditional variance $\boldsymbol{\lambda}_{t|t-1}$ can be estimated recursively in the minimum mean square error (MMSE) sense as follows

$$\begin{aligned}
\hat{\boldsymbol{\lambda}}_{t|t-1} &= E\left\{\boldsymbol{\lambda}_{t|t-1}|\mathbf{y}_0^{t-1};\boldsymbol{\theta}\right\} \\
&= E\left\{c_0\mathbf{1} + a_1\mathbf{x}_{t-1} \odot \mathbf{x}_{t-1}^* + b_1\boldsymbol{\lambda}_{t-1|t-2}|\mathbf{y}_0^{t-1};\boldsymbol{\theta}\right\} \\
&= c_0\mathbf{1} + a_1E\left\{\mathbf{x}_{t-1} \odot \mathbf{x}_{t-1}^*|\mathbf{y}_0^{t-1};\boldsymbol{\theta}\right\} + b_1E\left\{\boldsymbol{\lambda}_{t-1|t-2}|\mathbf{y}_0^{t-1};\boldsymbol{\theta}\right\} \\
&= c_0\mathbf{1} + a_1E\left\{\mathbf{x}_{t-1} \odot \mathbf{x}_{t-1}^*|\mathbf{y}_0^{t-1};\boldsymbol{\theta}\right\} + b_1\hat{\boldsymbol{\lambda}}_{t-1|t-2} \\
&= c_0\mathbf{1} + a_1E\left\{\mathbf{x}_{t-1} \odot \mathbf{x}_{t-1}^*|\mathbf{y}_{t-1},\hat{\boldsymbol{\lambda}}_{t-1|t-2};\boldsymbol{\theta}\right\} + b_1\hat{\boldsymbol{\lambda}}_{t-1|t-2}.
\end{aligned} \tag{5.12}$$

In order to compute $E\left\{\mathbf{x}_{t-1} \odot \mathbf{x}_{t-1}^*|\mathbf{y}_{t-1},\hat{\boldsymbol{\lambda}}_{t-1|t-2};\boldsymbol{\theta}\right\}$ note that this is the estimation of the *a posteriori* second order moment of vector $\mathbf{x}_{t-1}$ with *a priori* known covariance i.e. $\hat{R}_x^{t-1} = \mathrm{diag}(\hat{\boldsymbol{\lambda}}_{t-1|t-2})$. Using the fact that $\mathbf{x}_{t-1}$ is a zero mean complex Gaussian random vector corrupted with a Gaussian noise we have [[4] eq.14]

$$\begin{aligned}
\mathbf{u}_{t-1} &= E\left\{\mathbf{x}_{t-1} \odot \mathbf{x}_{t-1}^*|\mathbf{y}_{t-1},\hat{\boldsymbol{\lambda}}_{t-1|t-2};\boldsymbol{\theta}\right\} \\
&= \hat{R}_x^{t-1}\left(\hat{R}_x^{t-1} + \sigma^2\mathbf{I}\right)^{-1}\left[\mathrm{diag}\left(\sigma^2\mathbf{I}\right) + \hat{R}_x^{t-1}\left(\hat{R}_x^{t-1} + \sigma^2\mathbf{I}\right)^{-1}\mathbf{y}_{t-1} \odot \mathbf{y}_{t-1}^*\right].
\end{aligned} \tag{5.13}$$

In order to compute (5.9) note that

$$(\mathbf{x}_t|\mathbf{y}_0^{t-1};\boldsymbol{\theta}) \sim \mathcal{CN}\left(0,\mathrm{diag}(\hat{\boldsymbol{\lambda}}_{t|t-1})\right) \tag{5.14}$$

$$(\mathbf{n}_t|\mathbf{y}_0^{t-1};\boldsymbol{\theta}) \sim \mathcal{CN}\left(0,\sigma^2\mathbf{I}\right) \tag{5.15}$$

$$(\mathbf{y}_t=\mathbf{x}_t+\mathbf{n}_t|\mathbf{y}_0^{t-1};\boldsymbol{\theta}) \sim \mathcal{CN}\left(0,\sigma^2\mathbf{I}+\mathrm{diag}(\hat{\boldsymbol{\lambda}}_{t|t-1})\right). \tag{5.16}$$

From (5.9) through (5.16), the log-likelihood function of the current observation conditioned on past observations, i.e. $f(\mathbf{y}_t|\mathbf{y}_0^{t-1};\boldsymbol{\theta})$, can be written as

$$\log\left[f\left(\mathbf{y}_t|\mathbf{y}_0^{t-1};\boldsymbol{\theta}\right)\right] = \log\left[\frac{1}{\pi^k|\hat{R}_x^t+\sigma^2\mathbf{I}|}\right]-\mathbf{y}_t^*\left(\hat{R}_x^t+\sigma^2\mathbf{I}\right)^{-1}\mathbf{y}_t \tag{5.17}$$

where $|\cdot|$ is the determinant operation. The log-likelihood function of the observation can be computed using (5.9) and (5.17). To find the maximum of the log-likelihood function of the observation we must find the gradient and the Hessian matrix of this function.

The gradient of this function can be calculated by using

$$\begin{aligned}\mathbf{g}=\nabla\log\left[f\left(\mathbf{y}_0^{N-1};\boldsymbol{\theta}\right)\right] &= \nabla\sum_{t=1}^{N-1}\log\left[f\left(\mathbf{y}_t|\mathbf{y}_0^{t-1};\boldsymbol{\theta}\right)\right]\\ &= \sum_{t=1}^{N-1}\nabla\log\left[f\left(\mathbf{y}_t|\mathbf{y}_0^{t-1};\boldsymbol{\theta}\right)\right].\end{aligned} \tag{5.18}$$

In order to compute the gradient of the log-likelihood function of the current observation conditioned on past observations i.e. $\nabla\log\left[f\left(\mathbf{y}_t|\mathbf{y}_0^{t-1};\boldsymbol{\theta}\right)\right]$, Note that

$$\begin{aligned}\log\left[\frac{1}{\pi^k\left|\hat{R}_x^t+\sigma^2\mathbf{I}\right|}\right] &= -\log\pi^k-\log\left[\left|\hat{R}_x^t+\sigma^2\mathbf{I}\right|\right]\\ &= -\log\pi^k-\log\left[\prod_{i=1}^{k}\left(\hat{\lambda}_{i,t|t-1}+\sigma^2\right)\right]\\ &= -\log\pi^k-\sum_{i=1}^{k}\log\left[\hat{\lambda}_{i,t|t-1}+\sigma^2\right]\end{aligned} \tag{5.19}$$

and

$$\mathbf{y}_t^*\left(\hat{R}_x^t+\sigma^2\mathbf{I}\right)^{-1}\mathbf{y}_t=\sum_{i=1}^{k}\frac{y_{i,t}y_{i,t}^*}{\hat{\lambda}_{i,t|t-1}+\sigma^2} \tag{5.20}$$

where $y_{i,t}$ and $\hat{\lambda}_{i,t|t-1}$ are the i-th elements of $\mathbf{y}_t$, and $\hat{\boldsymbol{\lambda}}_{t|t-1}$ respectively. The gradient of the log-likelihood function of the current observation conditioned on past observations, can be com-

puted as follows

$$\frac{\partial}{\partial c_0}\log\left[f\left(\mathbf{y}_t|\mathbf{y}_0^{t-1};\boldsymbol{\theta}\right)\right] = -\sum_{i=1}^{k}\frac{\hat{\lambda}_{i,t|t-1}+\sigma^2-y_{i,t}y_{i,t}^*}{\left(\hat{\lambda}_{i,t|t-1}+\sigma^2\right)^2} \tag{5.21}$$

$$\frac{\partial}{\partial a_1}\log\left[f\left(\mathbf{y}_t|\mathbf{y}_0^{t-1};\boldsymbol{\theta}\right)\right] = -\sum_{i=1}^{k}\frac{\hat{\lambda}_{i,t|t-1}+\sigma^2-y_{i,t}y_{i,t}^*}{\left(\hat{\lambda}_{i,t|t-1}+\sigma^2\right)^2}u_{i,t-1} \tag{5.22}$$

$$\frac{\partial}{\partial b_1}\log\left[f\left(\mathbf{y}_t|\mathbf{y}_0^{t-1};\boldsymbol{\theta}\right)\right] = -\sum_{i=1}^{k}\frac{\hat{\lambda}_{i,t|t-1}+\sigma^2-y_{i,t}y_{i,t}^*}{\left(\hat{\lambda}_{i,t|t-1}+\sigma^2\right)^2}\hat{\lambda}_{i,t-1|t-2} \tag{5.23}$$

$$\frac{\partial}{\partial \sigma^2}\log\left[f\left(\mathbf{y}_t|\mathbf{y}_0^{t-1};\boldsymbol{\theta}\right)\right] = -\sum_{i=1}^{k}\frac{\hat{\lambda}_{i,t|t-1}+\sigma^2-y_{i,t}y_{i,t}^*}{\left(\hat{\lambda}_{i,t|t-1}+\sigma^2\right)^2}\left(1+a_1\frac{\partial u_{i,t-1}}{\partial \sigma^2}\right) \tag{5.24}$$

where $u_{i,t-1}$ is the i-th element of $\mathbf{u}_{t-1}$. $\hat{\lambda}_{i,t-1|t-2}$ and $u_{i,t-1}$ can be evaluated by (6.9) and (6.10), respectively, and the partial derivative of $u_{i,t-1}$ with respect to σ^2 can be computed as follows

$$\frac{\partial u_{i,t-1}}{\partial \sigma^2} = \frac{\hat{\lambda}_{i,t-1|t-2}^2\left(\hat{\lambda}_{i,t-1|t-2}+\sigma^2-2y_{i,t-1}y_{i,t-1}^*\right)}{\left(\hat{\lambda}_{i,t-1|t-2}+\sigma^2\right)^3}. \tag{5.25}$$

The Hessian matrix can be computed as follows

$$\begin{aligned}\mathbf{H}=\nabla^2\log\left[f\left(\mathbf{y}_0^{N-1};\boldsymbol{\theta}\right)\right] &= \nabla^2\sum_{t=1}^{N-1}\log\left[f\left(\mathbf{y}_t|\mathbf{y}_0^{t-1};\boldsymbol{\theta}\right)\right]\\ &= \sum_{t=1}^{N-1}\nabla^2\log\left[f\left(\mathbf{y}_t|\mathbf{y}_0^{t-1};\boldsymbol{\theta}\right)\right]\\ &= \sum_{t=1}^{N-1}\nabla^2\mathbf{H}_t\end{aligned} \tag{5.26}$$

where the second partial derivative of the log-likelihood function of the current observation conditioned on the past observations, i.e. $\mathbf{H}_t=\nabla^2\log\left[f\left(\mathbf{y}_t|\mathbf{y}_0^{t-1};\boldsymbol{\theta}\right)\right]$, can be computed as follows

$$\mathbf{H}_t=\begin{bmatrix}\frac{\partial^2(\gamma)}{\partial c_0^2} & \frac{\partial^2(\gamma)}{\partial c_0\partial a_1} & \frac{\partial^2(\gamma)}{\partial c_0\partial b_1} & \frac{\partial^2(\gamma)}{\partial c_0\sigma^2}\\ \frac{\partial^2(\gamma)}{\partial a_1\partial c_0} & \frac{\partial^2(\gamma)}{\partial a_1^2} & \frac{\partial^2(\gamma)}{\partial a_1\partial b_1} & \frac{\partial^2(\gamma)}{\partial a_1\sigma^2}\\ \frac{\partial^2(\gamma)}{\partial b_1\partial c_0} & \frac{\partial^2(\gamma)}{\partial b_1\partial a_1} & \frac{\partial^2(\gamma)}{\partial b_1^2} & \frac{\partial^2(\gamma)}{\partial b_1\partial\sigma^2}\\ \frac{\partial^2(\gamma)}{\partial\sigma^2\partial(c_0)^2} & \frac{\partial^2(\gamma)}{\partial\sigma^2\partial a_1} & \frac{\partial^2(\gamma)}{\partial\sigma^2\partial b_1} & \frac{\partial^2(\gamma)}{\partial\left(\sigma^2\right)^2}\end{bmatrix} \tag{5.27}$$

where

$$\gamma=\log\left[f\left(\mathbf{y}_t|\mathbf{y}_0^{t-1};\boldsymbol{\theta}\right)\right] \tag{5.28}$$

$$\frac{\partial^2(\gamma)}{\partial\left(c_0^2\right)^2} = \sum_{i=1}^{k}\frac{\hat{\lambda}_{i,t|t-1}+\sigma^2-2y_{i,t}y_{i,t}^*}{\left(\hat{\lambda}_{i,t|t-1}+\sigma^2\right)^3} \tag{5.29}$$

$$\frac{\partial^2(\gamma)}{\partial c_0\partial a_1} = \frac{\partial^2(\gamma)}{\partial a_1\partial c_0} = \sum_{i=1}^{k}\frac{\hat{\lambda}_{i,t|t-1}+\sigma^2-2y_{i,t}y_{i,t}^*}{\left(\hat{\lambda}_{i,t|t-1}+\sigma^2\right)^3}u_{i,t-1} \tag{5.30}$$

$$\frac{\partial^2(\gamma)}{\partial c_0\partial b_1} = \frac{\partial^2(\gamma)}{\partial b_1\partial c_0} = \sum_{i=1}^{k}\frac{\hat{\lambda}_{i,t|t-1}+\sigma^2-2y_{i,t}y_{i,t}^*}{\left(\hat{\lambda}_{i,t|t-1}+\sigma^2\right)^3}\hat{\lambda}_{i,t-1|t-2} \tag{5.31}$$

$$\frac{\partial^2(\gamma)}{\partial c_0\partial\sigma^2} = \frac{\partial^2(\gamma)}{\partial\sigma^2\partial c_0} = \sum_{i=1}^{k}\frac{\hat{\lambda}_{i,t|t-1}+\sigma^2-2y_{i,t}y_{i,t}^*}{\left(\hat{\lambda}_{i,t|t-1}+\sigma^2\right)^3}\left(1+a_1\frac{\partial u_{i,t-1}}{\partial\sigma^2}\right) \tag{5.32}$$

$$\frac{\partial^2(\gamma)}{\partial a_1^2} = \sum_{i=1}^{k}\frac{\hat{\lambda}_{i,t|t-1}+\sigma^2-2y_{i,t}y_{i,t}^*}{\left(\hat{\lambda}_{i,t|t-1}+\sigma^2\right)^3}u_{i,t-1}^2 \tag{5.33}$$

$$\begin{aligned}\frac{\partial^2(\gamma)}{\partial a_1\partial b_1} &= \frac{\partial^2(\gamma)}{\partial b_1\partial a_1}\\ &= \sum_{i=1}^{k}\frac{\hat{\lambda}_{i,t|t-1}+\sigma^2-2y_{i,t}y_{i,t}^*}{\left(\hat{\lambda}_{i,t|t-1}+\sigma^2\right)^3}\hat{\lambda}_{i,t-1|t-2}u_{i,t-1}\end{aligned} \tag{5.34}$$

$$\begin{aligned}\frac{\partial^2(\gamma)}{\partial a_1\partial\sigma^2} = \frac{\partial^2(\gamma)}{\partial\sigma^2\partial a_1} &= \sum_{i=1}^{k}\frac{\hat{\lambda}_{i,t|t-1}+\sigma^2-2y_{i,t}y_{i,t}^*}{\left(\hat{\lambda}_{i,t|t-1}+\sigma^2\right)^3}\left(1+a_1\frac{\partial u_{i,t-1}}{\partial\sigma^2}\right)u_{i,t-1}\\ &\quad-\sum_{i=1}^{k}\frac{\partial u_{i,t-1}}{\partial\sigma^2}\frac{\hat{\lambda}_{i,t|t-1}+\sigma^2-y_{i,t}y_{i,t}^*}{\left(\hat{\lambda}_{i,t|t-1}+\sigma^2\right)^2}\end{aligned} \tag{5.35}$$

$$\begin{aligned}\frac{\partial^2(\gamma)}{\partial b_1\partial\sigma^2} &= \frac{\partial^2(\gamma)}{\partial\sigma^2\partial b_1}\\ &= \sum_{i=1}^{k}\frac{\hat{\lambda}_{i,t|t-1}+\sigma^2-2y_{i,t}y_{i,t}^*}{\left(\hat{\lambda}_{i,t|t-1}+\sigma^2\right)^3}\left(1+a_1\frac{\partial u_{i,t-1}}{\partial\sigma^2}\right)\hat{\lambda}_{i,t-1|t-2}\end{aligned} \tag{5.36}$$

$$\begin{aligned}\frac{\partial^2(\gamma)}{\partial(\sigma^2)^2} &= \sum_{i=1}^{k}\frac{\hat{\lambda}_{i,t|t-1}+\sigma^2-2y_{i,t}y_{i,t}^*}{\left(\hat{\lambda}_{i,t|t-1}+\sigma^2\right)^3}\left(1+a_1\frac{\partial u_{i,t-1}}{\partial\sigma^2}\right)\\ &\quad-a_1\sum_{i=1}^{k}\frac{\hat{\lambda}_{i,t|t-1}+\sigma^2-y_{i,t}y_{i,t}^*}{\left(\hat{\lambda}_{i,t|t-1}+\sigma^2\right)^2}\frac{\partial^2 u_{i,t-1}}{\partial\left(\sigma^2\right)^2}\end{aligned} \tag{5.37}$$

Table 5.1: Algorithm For ML Parameter Estimation of Noisy Complex GARCH Model Without Considering Stationarity Conditions

Initialization:
Choose $\boldsymbol{\theta}_0$;
Let $\mathbf{u}_1 = \boldsymbol{\lambda}_{1\|0} = \mathbf{0}_{k\times 1}$ and $I = 0$;
while $\mathbf{g} > 0$
Compute $\mathbf{u}_t$ for $t = 1,2,...,N-1$ using (6.10);
Compute $\hat{\boldsymbol{\lambda}}_{t\|t-1}$ for $t = 1,2,...,N-1$ using (6.9);
Compute $\mathbf{g}$ and $\mathbf{H}$ using (5.18) and (6.36) respectively.
$\boldsymbol{\theta}_{I+1} = \boldsymbol{\theta}_I + \gamma \mathbf{H}^{-1}\mathbf{g}$;
$I = I + 1$;
end (while)

$$\frac{\partial^2 u_{i,t-1}}{\partial (\sigma^2)^2} = -2\frac{\hat{\lambda}^2_{i,t-1|t-2}\left(\hat{\lambda}_{i,t-1|t-2} + \sigma^2 - 3y_{i,t-1}y^*_{i,t-1}\right)}{\left(\hat{\lambda}_{i,t-1|t-2} + \sigma^2\right)^4}. \tag{5.38}$$

With these values of the gradient vector and Hessian matrix, we can use numerical optimization methods such as Newton [76] or gradient projection method [76] to find the maximum of the log-likelihood function with or without considering stationarity conditions. The algorithm for estimating the parameters without considering the stationarity conditions is summarized in Table I.

In order to introduce the algorithm for estimating the parameters under the stationarity conditions, note that the constraints in (5.8) and positiveness of σ^2 can be written as $\mathbf{M}\boldsymbol{\theta} \leq \boldsymbol{\mu}$ where $\mathbf{M} = \left[-\mathbf{I}_{4\times 4}, [0,1,1,0]^T\right]^T$, $\boldsymbol{\mu} = [0,0,0,0,1]^T$ and $\mathbf{a} \leq \mathbf{b}$ means that each element of the vector $\mathbf{a}$ is less than or equal to the corresponding element in the vector $\mathbf{b}$. Our final algorithm is summarized in Table II. In this algorithm, $\mathbf{M}_q$ is constructed from $\mathbf{M}$ by eliminating the rows corresponding to inactive constraints.

It is worth mentioning that if the covariance matrix of the additive noise is known, i.e. the signal to noise ratio (SNR) is known *a priori*, the proposed method can be used in the same manner by eliminating the last element of the gradient vector and the last column and the last row of the Hessian matrix.

Table 5.2: Algorithm for ML Parameter Estimation of Noisy Complex GARCH Model, Under Stationarity Conditions

Initialization:

Chose $\boldsymbol{\theta}_0$ which satisfies the stationarity conditions

Let $\mathbf{u}_1 = \hat{\boldsymbol{\lambda}}_{1|0} = 0$;

1) Find the subspace of active constraints and form $\mathbf{M}_q$.

2) Calculate $\mathbf{P} = \mathbf{I} - \mathbf{M}_q^T \left(\mathbf{M}_q \mathbf{M}_q^T\right)^{-1} \mathbf{M}_q$, and $\mathbf{d} = \mathbf{P}\mathbf{g}$ using (5.18) through (5.25).

3) If $\mathbf{d} \neq \mathbf{0}$, find ρ_1 and ρ_2 satisfying

$\max \{\rho_1 : \boldsymbol{\theta} + \rho_1 \mathbf{d} \text{ is feasible}\}$

$\max \left\{\log\left(f\left(y_1^N; \boldsymbol{\theta} + \rho_2 \mathbf{d}\right)\right) : 0 \leq \rho_2 \leq \rho_1\right\}$

using (5.9). Set $\boldsymbol{\theta}$ to $\boldsymbol{\theta} + \rho_2 \mathbf{d}$

5) If $\mathbf{d} = 0$, find $\boldsymbol{\lambda} = \left(\mathbf{M}_q \mathbf{M}_q^T\right)^{-1} \mathbf{M}_q \mathbf{g}$.

a) If $\boldsymbol{\lambda}_j \geq 0$ for all j corresponding to active inequalities, stop; $\boldsymbol{\theta}$ satisfies the Kuhn-Tucker conditions.

b) Otherwise, delete the row from $\mathbf{M}_q$, corresponding to the inequality with the most negative component of $\boldsymbol{\lambda}$ and return to 2.

Another issue that must be addressed here is the case of additive colored noise. Suppose that $n_t \sim \mathcal{CN}(0, \boldsymbol{\Sigma}_n)$ where $\boldsymbol{\Sigma}_n$ is a *known* matrix with σ_i^2 on its diagonal. The off-diagonal entries are assumed to be zero as in all noise modeling procedures in the STFT domain for speech signal processing. In this case the terms $\sigma^2 \mathbf{I}$ and $\hat{\lambda}_{i,t|t-1} + \sigma^2$ in the above formulas must be replaced by $\boldsymbol{\Sigma}_n$ and $\hat{\lambda}_{i,t|t-1} + \sigma_i^2$ respectively.

5.3.1 Recursive ML

A standard approach to Recursive ML (RML) estimation considers a series of log-likelihood functions $\log\left[f\left(\mathbf{y}_0^k; \boldsymbol{\theta}\right)\right]$, where $\log\left[f\left(\mathbf{y}_0^k; \boldsymbol{\theta}\right)\right] = \sum_{t=1}^k \log\left[f\left(\mathbf{y}^t | \mathbf{y}_0^{t-1}; \boldsymbol{\theta}\right)\right]$ [75]. Under suitable regularity conditions described in [117] it can be shown that the average log-likelihood

converges to the following limit

$$l(\boldsymbol{\theta}) = \lim_{k\to\infty} \frac{1}{k+1} \sum_{t=1}^{k} \log\left[f\left(\mathbf{y}^t|\mathbf{y}_0^{t-1};\boldsymbol{\theta}\right)\right]. \tag{5.39}$$

It can be shown that $l(\boldsymbol{\theta})$ admits $\boldsymbol{\theta}^*$ as a global maximum where $\boldsymbol{\theta}^*$ is the global maximum of the log-likelihood function [75]. The function $l(\boldsymbol{\theta})$ does not have an analytical expression and we do not have access to it. Nevertheless, identification of $\boldsymbol{\theta}^*$ can still be achieved based on the ergodicity property in (6.35), which provides us with a set of accessible functions $f\left(\mathbf{y}^t|\mathbf{y}_0^{t-1};\boldsymbol{\theta}\right)$ that converge to $l(\boldsymbol{\theta})$. One way to exploit this in order to maximize $l(\boldsymbol{\theta})$, is to use a Stochastic Approximation (SA) algorithm to update the parameter estimate at time n using the recursion

$$\boldsymbol{\theta}_n = \boldsymbol{\theta}_{n-1} + \gamma_n \nabla \log\left[f\left(\mathbf{y}_n|\mathbf{y}_0^{n-1};\boldsymbol{\theta}\right)\right] \tag{5.40}$$

where $\boldsymbol{\theta}_{n-1}$ is the parameter estimate at time $n-1$ and $\nabla \log\left[f\left(\mathbf{y}_n|\mathbf{y}_0^{n-1};\boldsymbol{\theta}\right)\right]$ can be computed by (6.13)–(6.15), provided that the step size γ_n is a positive non-increasing sequence, such that $\sum_{n=1}^{\infty}\gamma_n = \infty$ and $\sum_{n=1}^{\infty}\gamma_n^2 < \infty$. It can be shown that $\boldsymbol{\theta}_n$ will converge to the set of (global or local) maxima of $l(\boldsymbol{\theta})$. It is worth mentioning that this method (RML) can be used for adaptive parameter estimation (varying parameters). In order to achieve this goal one should use the RML with constant step size (i.e., $\gamma_n = \gamma$). The choice of γ will be a trade-of between tracking capability (large γ) and low estimation noise around the parameter (small γ).

5.4 State smoothing

After the estimation of the parameters of the complex GARCH model these estimated parameters can be used to estimate the hidden state of the model, $\mathbf{x}_t$, in the MMSE sense. In order to do this, note that the MMSE optimal estimator in a Bayesian framework is the conditional mean

$$\hat{\mathbf{x}}_t = E\left\{\mathbf{x}_t|\mathbf{y}_0^t\right\} = E\left\{\mathbf{x}_t|\hat{\boldsymbol{\lambda}}_{t|t-1},\mathbf{y}_t\right\}. \tag{5.41}$$

Because $\mathbf{x}_t$ conditioned on $\hat{\boldsymbol{\lambda}}_{t|t-1}$ and $\mathbf{y}_t$ is a zero mean complex Gaussian vector with covariance matrix equals to diag $\left(\hat{\boldsymbol{\lambda}}_{t|t-1}\right)$, the solution of the conditional expectation in (6.23) is the well-known Wiener filter. Hence, the conditional mean in (6.23) can be calculated using (6.9), (6.10), the estimated parameters (i.e. $\hat{\boldsymbol{\theta}} = [\hat{c}_0, \hat{a}_1, \hat{b}_1, \hat{\sigma}^2]^T$), and the following equation

$$\hat{\mathbf{x}}_t = \hat{R}_x^t\left(\hat{R}_x^t + \hat{\sigma}^2\mathbf{I}\right)^{-1}\mathbf{y}_t. \tag{5.42}$$

5.5 Simulations

In this section, we analyze the performance of our proposed method using three different experiments. In the first experiment, we use a two dimensional $(k = 2)$ GARCH$(1,1)$ process corrupted with zero-mean complex Gaussian white noise with three different signal to noise ratios (SNR i.e. -10, -5 and 0 dB). The number of available data was set to $N = 1000$ samples. The process $\mathbf{v}_t$ is zero-mean Gaussian white process with identity covariance matrix. The parameters of the GARCH$(1,1)$ are $\boldsymbol{\theta} = [0.10, 0.40, 0.30]^T$. We compare the performance of the proposed method with that of two different ML methods. The first method denoted by MLClean, employs the clean data (unavailable in practical situations) for estimating the parameters. The second method denoted by MLNoisy, employs the noisy data for estimating the parameters assuming that the data is clean. The proposed method is examined under two different situations. The first situation denoted by PUN, employs the noisy data for estimating the parameters assuming that the corrupting noise variance σ^2 is unknown. The second situation denoted by PKN, employs noisy data for estimating the parameter assuming that the corrupting noise variance is known.

For evaluating the performance of our method we used the mean square error (MSE) in parameter estimation and the output SNR of the proposed state smoothing algorithm which both are estimated using 2000 realizations. The input SNR (SNR_{in}) and the output SNR (SNR_{out}) are defined as follows

$$\begin{aligned}
\text{SNR}_{\text{in}} &= 10\log_{10}\frac{1}{2000}\sum_{i=1}^{2000}\frac{\sum_{t=1}^{N}\left\|\mathbf{x}_t^i\right\|^2}{\sum_{t=1}^{N}\left\|\mathbf{x}_t^i-\mathbf{y}_t^i\right\|^2} \\
\text{SNR}_{\text{out}} &= 10\log_{10}\frac{1}{2000}\sum_{i=1}^{2000}\frac{\sum_{t=1}^{N}\left\|\mathbf{x}_t^i\right\|^2}{\sum_{t=1}^{N}\left\|\mathbf{x}_t^i-\hat{\mathbf{x}}_t^i\right\|^2}
\end{aligned} \tag{5.43}$$

where $\mathbf{x}_t^i$, $\mathbf{x}_t^i$ and $\hat{\mathbf{x}}_t^i$ are the clean, noisy and estimated signals in the i-th realization, respectively. For each previously mentioned parameter estimation methods $\hat{\mathbf{x}}_t^i$ is obtained using the proposed state smoothing algorithm assuming that the corrupting noise variance is known. The results are depicted in Tables 5.3 to 5.5. In the last columns of these tables we also provide the output SNR for minimum mean square error estimator (MMSEE). This estimator uses the true one-frame-ahead conditional variance of the clean signal $(\boldsymbol{\lambda}_{t|t-1})$ for state smoothing (i.e. the MMSEE estimator uses $(\boldsymbol{\lambda}_{t|t-1})$ instead of its estimate $(\hat{\boldsymbol{\lambda}}_{t|t-1})$ in (5.42)). Note that the output SNR

Table 5.3: MSE in Parameter Estimation for Different Methods (SNR_{in}=-10dB)

MSE	MLClean	MLNoisy	PKN	PUN	MMSEE
c_0	0.0003	13.9474	0.0196	7.0087	—
a_1	0.0010	0.1506	0.1161	0.1385	—
b_1	0.0024	0.1258	0.0530	0.0901	—
SNR_{out}	0.3656	-4.8379	0.3357	-3.5234	0.4991

Table 5.4: MSE in Parameter Estimation for Different Methods (SNR_{in}=-5dB)

MSE	MLClean	MLNoisy	PKN	PUN	MMSEE
c_0	0.0003	0.8057	0.0051	0.3590	—
a_1	0.0010	0.1314	0.0628	0.0881	—
b_1	0.0024	0.1595	0.0830	0.0931	—
SNR_{out}	1.2229	-0.7952	1.1940	0.1835	1.4405

given in this column is the highest achievable SNR. It is obvious that the proposed methods yield better performance in output SNR and parameter estimation over the MLNoisy method which is often used in real world problems. The best performance is obtained by using the MLClean, as expected, but this method cannot be used in practical situations because the clean data is not available. Comparing the output SNR of PKN and MMSEE shows that the performance of the proposed parameter estimation method is very close to the best achievable performance (i.e. MMSEE). Another point which must be emphasized here is that the performance of the proposed method degrades if the noise variance is unknown. This is because of the additional information employed in PKN method. Another issue which must be mentioned here is that estimating b_1 is much more difficult than c_0 and a_1 because b_1 is multiplied by a hidden quantity $\boldsymbol{\lambda}_{t|t-1}$ which makes the estimation of this parameter difficult. The simulation results confirm this fact. Other simulation results show that the performance of MLClean, MLNoisy, PUN and PKN methods are almost the same for high values of SNR (grater than 5 dB).

In the second experiment we investigate the influence of the colored noise on the proposed algorithm. In the text we addressed how to treat the colored noise. For this experiment we used the same GARCH model as expressed in the first simulation but the process was corrupted by a

Table 5.5: MSE in Parameter Estimation for Different Methods (SNR_{in}=0.0dB)

MSE	MLClean	MLNoisy	PKN	PUN	MMSEE
c_0	0.0003	0.0316	0.0009	0.0223	—
a_1	0.0010	0.0740	0.0171	0.0513	—
b_1	0.0024	0.0401	0.0278	0.0387	—
SNR_{out}	3.1225	2.6878	3.1128	2.8427	3.3009

Table 5.6: MSE in Parameter Estimation for Different Methods (Colored Noise) (SNR_{in}=-4.44dB)

MSE	MLClean	MLNoisy	PKN	MMSEE
c_0	0.0003	0.0034	0.0014	—
a_1	0.0020	0.0529	0.0145	—
b_1	0.0047	0.2358	0.0244	—
SNR_{out}	1.5935	0.9356	1.5604	1.7905

zero-mean Gaussian colored noise with covariance matrix equals to $\Sigma_n = \text{diag}\,(1, 0.16)$. By this choice of the corrupting noise covariance matrix the input SNR equals to -4.44 dB. The results are depicted in Table 5.6. It is apparent that the proposed method outperforms the MLNoisy method.

In the third experiment, we try to estimate the time varying parameters of a two dimensional complex GARCH process corrupted with zero-mean complex Gaussian white process with known covariance using the proposed RML method. The performance of the RML method is analyzed in three different SNRs i.e. 0, 6 and 10 dB. The true and estimated values of the time varying parameters are depicted in Fig.1. It is obvious that the RML algorithm can follow time varying parameters and its performance gets better as the SNR increases. The MSE in parameter estimation shown in Table 5.7, is estimated by

$$\text{MSE}_i = \frac{1}{10^4} \sum_{t=1}^{10^4} \left(\theta_i(t) - \hat{\theta}_i(t)\right)^2 \tag{5.44}$$

where θ_i, $\hat{\theta}_i$ and MSE_i are i-th element of $\boldsymbol{\theta}$, its estimate and mean square error in its estimation respectively. For a practical application of the RML parameter estimation in speech processing see [84].

Table 5.7: MSE in Parameter Estimation for RML Methods with Three Different SNR

MSE	SNR=0 dB	SNR=6 dB	SNR=10 dB
c_0	0.0190	0.0075	0.0065
a_1	0.0245	0.0147	0.0100
b_1	0.0143	0.0138	0.0114

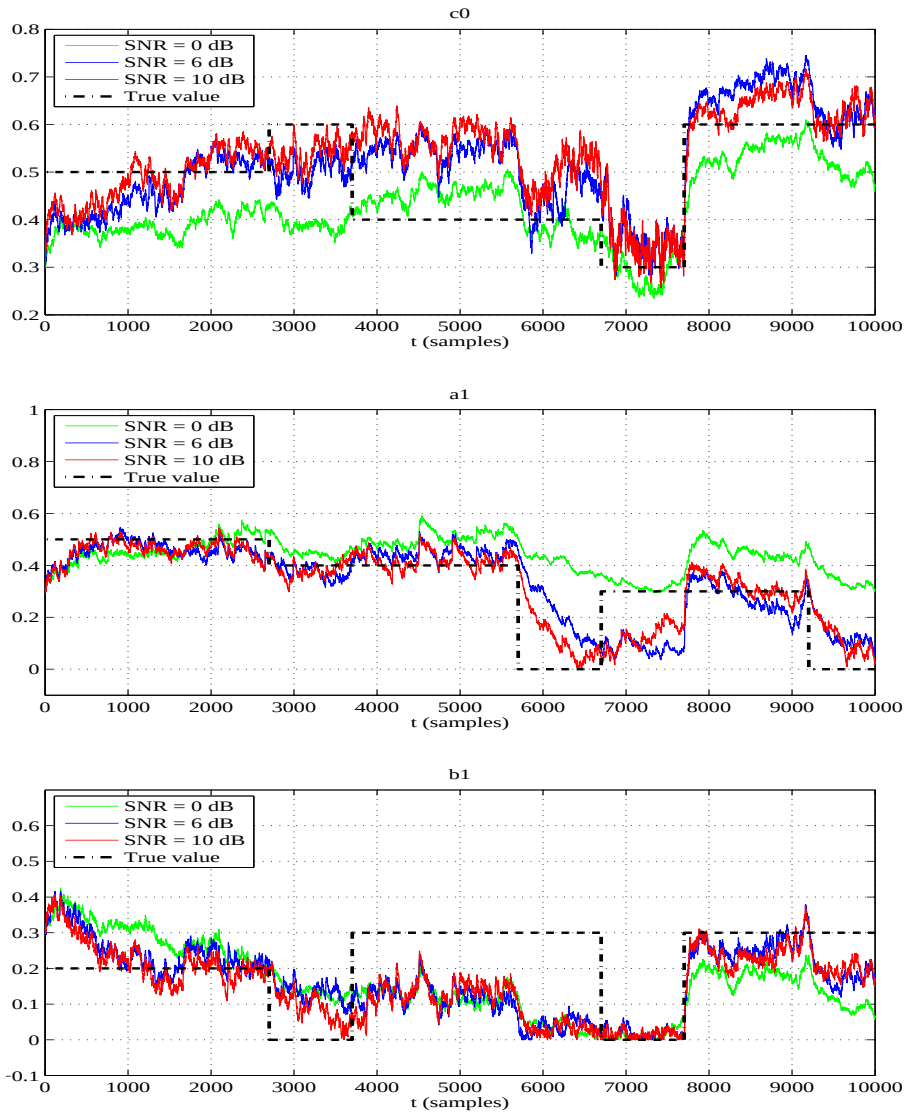


Figure 5.1: The results of the RML method in estimating time varying parameters for three different SNRs.

5.5.1 Computational Complexity

The computational complexities of the MLNoisy and the MLClean methods, which are often used in practice, are the same since these methods only use different data. The computational complexity of the proposed method roughly equals to that of MLNoisy (or MLClean) method. The reason is that all these methods are using the same algorithm i.e. gradient projection method for maximization of the pdf on the parameters. Assuming that multiplications and divisions have the same complexity for the processor, the number of sums and multiplications for computing the pdf function in each iteration of the gradient projection method for the MLClean method are $3kN$ and $6kN$ respectively. For the proposed method the number of sums is $6kN$ and the number of multiplications is $9kN$ respectively. Increasing the order of the model increases the computational load but the computational load of parameter estimation of higher order models is still $O(N)$. It should be mentioned that the method presented in [95] takes approximately 35 hours to estimate the parameters of one process while the proposed method just needs 23 seconds (for the process used in the first simulation). This fact shows that the method presented in [95], even if it has better performance over the purposed method, cannot be applied in practical applications.

5.6 Conclusion

We have presented a new ML estimation procedure for parameter estimation of complex GARCH$(1,1)$ model in presence of the additive noise, and a new procedure for estimating the hidden states of the model in the MMSE sense. The method computes the log-likelihood function together with its gradient vector and Hessian matrix, and uses the Newton method (or gradient projection method under stationarity conditions) to numerically find the maximum of the log-likelihood function. We also presented an RML method for estimating the time varying parameters. Simulation results show improvement with respect to not considering the noise effect in the estimation of the parameters. In this work we considered two situations. The first one is the general case where the SNR is unknown. In this situation, the proposed method estimates the variance of the noise together with the parameters and the hidden states of the model. The second situation, which is a special case of the first one, assumes that the variance of the additive noise is known and just estimates the parameters of the model. As expected, the

second case yields better performance over the first one because of the additional information employed. The issue how to handle colored noise is also discussed in the chapter.

Chapter 6

AR-GARCH in Presence of Noise: Parameter Estimation and its Application to Voice Activity Detection

Voice activity detection (VAD) is an integral part of all speech communication systems. Examples of such systems are audio conferencing, hands-free telephony [49] and discontinuous speech transmission [16]. During the last few decades, many researchers have dealt with this issue. The modern VADs are derived from statistical model-based algorithms in which the likelihood ratio test (LRT) is applied to a set of hypotheses. These VADs are developed in [3]-[6]. In all of these methods, the speech signal is transformed to the short-time Fourier transform (STFT) domain and different models are applied to the speech data in the STFT domain. For instance, Sohn et al. [111] and Ramirez et al. [96] assumed that the spectral coefficients of the noise and speech signal are complex gaussian random variables. Chang et al. [21] utilized the complex Laplacian and Gamma probability density functions (pdfs) to model the distributions of the speech and noise spectra. Shin et al. [108] applied the generalized gamma distribution to model the distribution of the clean speech spectrum. Davis et al. [37] proposed the statistical VAD method that makes no assumption about the distributions of the speech spectra.

Recently, Cohen [25] modeled the speech signal in the STFT domain as a complex GARCH process and used this model for speech enhancement. He showed that the time varying variance of the speech signal can be estimated using a complex GARCH model with Gaussian innovations. The well-known decision-directed method of Ephraim and Malah [43] can be derived as

a special case using GARCH modeling. Solvang et al. [112] used the AR-GARCH model for VAD. In their work, they represent the AR part of the AR-GARCH model with a state-space to obtain the appropriate linear prediction error series. By applying the GARCH model to the residuals, they estimate the conditional variance sequences corresponding to the voice activity parts. To detect voice activity, they establish an appropriate threshold for the conditional variance sequences. However, their method is not computationally efficient and cannot be used in real time speech signal processing. Furthermore, they assume that the model is identified from frames with speech being present, but in the application at hand these frames are unknown to the user in advance.

In this chapter, we present a new VAD using AR-GARCH modeling of the speech signal in the time domain. This model relies on the fact that speech signals in the time domain can be modeled as AR processes [15] and also demonstrate both the variability clustering and the heavy-tail behavior, that are the main properties of GARCH processes. Our VAD is based on the LRT, so we need to compute the likelihood function of the observations under the speech presence and absence hypotheses. The time varying variance of the speech signal needed for computing the likelihood function under speech presence hypothesis is estimated using the AR-GARCH model. To be able to use this model, one should know the parameters of the model or estimate them from the available data. Since the available data is often corrupted by additive noise, we present a novel technique for parameter estimation of the AR-GARCH model in presence of additive noise. To compute the likelihood function of the observations under speech presence and absence hypotheses, we need to know the variance of the additive noise, which might be time varying. The improved minima controlled recursive averaging (IMCRA) [23] method has been modified and utilized for estimating the variance of the corrupting noise in the time domain.

The chapter is organized as follows. In Section 8.2, we introduce the AR-GARCH model and show its applicability to speech signals. In Section 8.3, we introduce our parameter estimation algorithm. In Section 8.4, we introduce our VAD, which is based on the AR-GARCH modeling of the speech signal in the time domain. Finally in Section 6.4, we examine the performance of our method using several simulations.

6.1 AR versus AR-GARCH modeling of speech signals

The GARCH model was first introduced by Bollerslev [19] as an extension of the ARCH model developed by Engle [42] in economic data modeling. Since then, many researchers have tried to expand and use these models in several applications. AR-GARCH is one of these extensions [74]. The AR(p)-GARCH$(1,1)$ process is a filtered version of a GARCH$(1,1)$ process with an all-pole filter. This model is defined by the following three equations,

$$x_t = \sum_{i=1}^{p} \alpha_i x_{t-i} + \varepsilon_t \tag{6.1}$$

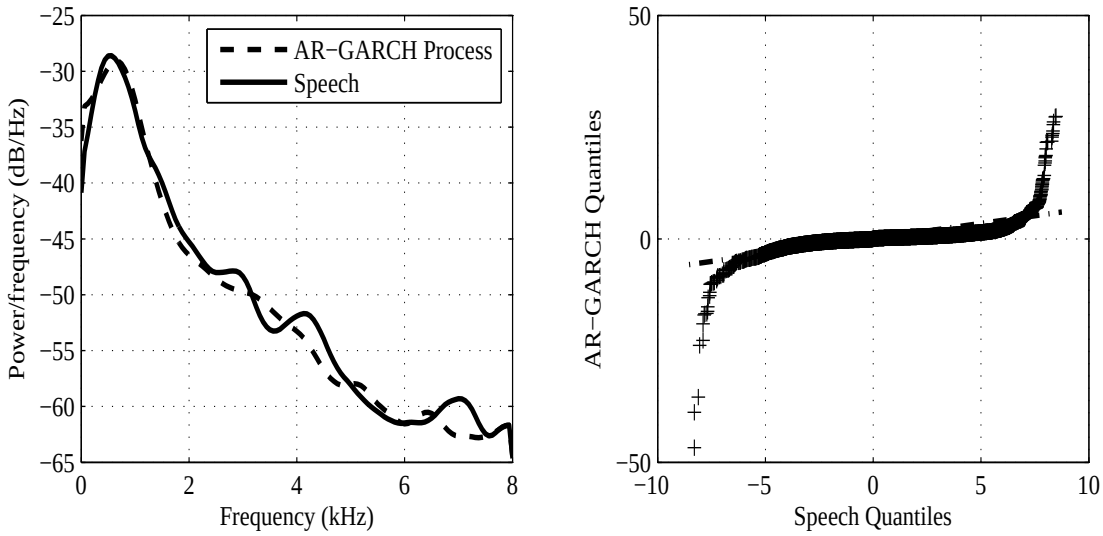
$$\varepsilon_t = \sigma_{t|t-1} v_t \tag{6.2}$$

$$\sigma^2_{t|t-1} = \beta_0 + \beta_1 \varepsilon^2_{t-1} + \beta_2 \sigma^2_{t-1|t-2} \tag{6.3}$$

where t is the time index, α_i's and p in (6.1) are the parameters and the order of the AR part respectively, v_t's in (6.2) are zero-mean independent identically distributed random variables with unity variance, β_i's in (6.3) are the parameters of the GARCH$(1,1)$ model, and $\sigma^2_{t|t-1}$ is the one-sample-ahead conditional variance of the clean signal. We assume that v_t's are Gaussian random variables because this distribution models the speech data better than the Laplace or Gamma distributions [25].

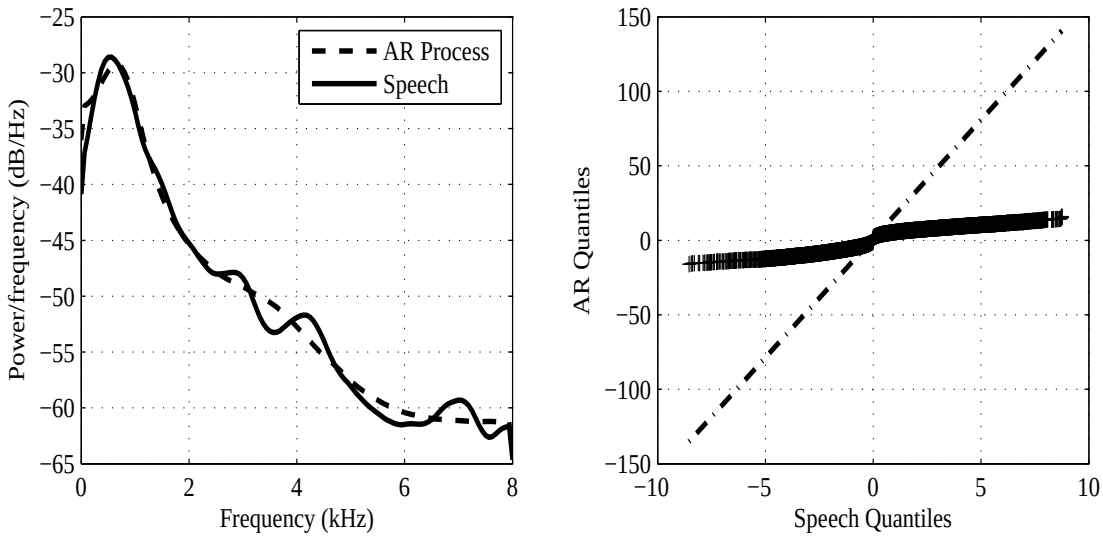
We use the AR-GARCH to model the speech signal in the time domain, because it can model two main characteristics of the speech signals. Practical evidence shows that the spectrum of speech signals is very similar to that of AR processes. Since the spectrum of AR-GARCH processes depends only on the coefficients of the AR part, we can use the AR-GARCH model to model the spectral characteristics of speech signals. Another common property of speech signals and AR-GARCH processes is that they both have heavy tail pdfs. Fig. 6.1 (a) shows the spectrum of a typical clean speech signal consisting of both male and female speakers from the TIMIT database [50], together with the spectrum of a synthetic AR(5)-GARCH$(1,1)$ process. The spectrums are estimated using the Welch method [115]. The synthetic AR-GARCH process is simulated using the parameters estimated from the clean speech signal using the RML method (discussed in the next section). From this figure, it is obvious that the spectrum of the speech signal is very similar to that of the synthetic AR-GARCH process. Fig. 6.1 (b) displays the quantile-quantile plot (QQ-plot) of the quantiles of the speech signal versus the quantiles of the synthetic AR-GARCH samples. The purpose of the QQ plot is to determine whether

the samples of two processes come from the same distribution. If the samples do come from the same distribution (same shape), even if one distribution is shifted and re-scaled from the other (different location and scale parameters), the plot will be linear. A reference line passing through the first and third quartiles is helpful for judging whether the points are linear. We provide in Fig. 6.2 (a) the spectrum of a synthetic AR(5) process with Gaussian residual samples together with the spectrum of a clean speech signal. In Fig. 6.2 (b), we also provide the QQ-plot of the quantiles of the speech signal versus the quantiles of the synthetic AR(5) process with Gaussian residual samples. From Figs. 6.1 through 6.2 , we see that although both AR and AR-GARCH models can model the spectral properties of speech signals very well, the QQ-plot of the quantiles of the speech signal versus the quantiles of the AR-GARCH process is linear in a wider range when comparing the quantiles of the speech signal versus the quantiles of the AR process. This means that an AR-GARCH model can model the speech signal in the time domain much better than the widely-used AR model.



(a) (b)

Figure 6.1: (a) Spectrum of a typical clean speech signal together with the spectrum of a synthetic AR-GARCH process. (b) QQ-plot of the quantiles of speech signal versus the quantiles of a synthetic AR-GARCH process.



(a) (b)

Figure 6.2: (a) Spectrum of a typical clean a speech signal together with the spectrum of a synthetic AR process. (b) QQ-plot of the quantiles of a speech signal versus the quantiles of a synthetic AR process with Gaussian residual.

6.2 AR-GARCH parameter estimation

In this section we introduce a computationally efficient method for estimating the parameters of the AR-GARCH model from noisy observations. This method is then used in the next section for obtaining the proposed VAD.

Let y_t be a corrupted version of an AR-GARCH process with additive white Gaussian noise, i.e.,

$$y_t = x_t + n_t \qquad t = 1,2,3,...,N \tag{6.4}$$

where N is the number of available data and n_t is the sequence of additive white Gaussian noise independent of x_t with unknown variance, i.e.,

$$n_t \sim \mathcal{N}\left(0, \sigma_t^2\right). \tag{6.5}$$

Our purpose is to estimate the hidden states (x_t's) and the parameters of the model $\boldsymbol{\theta} = [\beta_0, \beta_1, \beta_2, \ \alpha_1, \alpha_2, ..., \alpha_p]^T$ using the available noisy data $y_t; t = 1,2,3,...,N$.

The first step in obtaining the states and the parameter estimators is to estimate the variance of the corrupting noise (i.e., σ_t^2). Although most of the authors assume that the variance of the additive noise is known, in the current application (i.e., VAD) this assumption is not reasonable.

The statistical properties of the noise are estimated from the speech-free frames and in this application these frames are unknown to the user in advance. Furthermore, the variance of the additive noise can be time varying. Hence, we need a method for estimating it adaptively. The IMCRA method is used for estimating the statistical properties of the additive noise (i.e., its variance). Note that the IMCRA method estimates the variance of the noise in the frequency domain (the STFT domain) while we need the variance of the noise in the time domain. In order to estimate the variance of the noise in the time domain using the estimates of the variance of the noise in the STFT domain, we use the well-known Parseval's relation. Suppose that the m-th frame consists of K samples of noisy speech signal and assume that the noise is stationary in this frame. Using Parseval's relation, we have,

$$\begin{aligned}
\sum_{k=1}^{K} |N_m(k)|^2 &= K \sum_{t=(m-1)K+1}^{mK} n^2(t) \\
\sum_{k=1}^{K} E\left\{|N_m(k)|^2\right\} &= K \sum_{t=(m-1)K+1}^{mK} E\left\{n^2(t)\right\} \\
\sum_{k=1}^{K} \Gamma_m(k) &= K^2 \sigma_m^2 \qquad (6.6)
\end{aligned}$$

where $N_m(k)$ is the STFT coefficient of the noise in time frame number m and in the k-th frequency bin. $\Gamma_m(k)$ is the variance of the noise in the STFT domain in the m-th time frame and the k-th frequency bin. Thus, the estimate of the noise for a K sample time frame is as follows

$$\hat{\sigma}_t^2 = \frac{1}{K^2} \sum_{k=1}^{K} \hat{\Gamma}_m(k) \quad t = (m-1)K+1, 2, ..., mK \qquad (6.7)$$

where $\hat{\Gamma}_m(k)$ is the estimate of the variance of the noise in the STFT domain in the m-th time frame and the k-th frequency bin obtained by the IMCRA method.

Using this estimate of the noise variance, we now introduce our state smoothing and parameter estimation method. The optimal estimator in the MMSE sense is the conditional mean, i.e.,

$$\hat{x}_t = E\left\{x_t \left| y_1^t \right.\right\} \qquad (6.8)$$

where $y_1^t = \{y_\tau | \tau = 1, 2, ..., t\}$ is the set of observations up to time t. Let $x_1^t = \{x_\tau | \tau = 1, 2, ..., t\}$ and $\hat{x}_1^t = \{\hat{x}_\tau | \tau = 1, 2, ..., t\}$ be the set of clean samples (hidden states) and their estimates up to

time t, respectively. Furthermore, as in [25], we assume that past estimates of the one-sample-ahead conditional variance and past estimates of the clean signal are sufficient statistics for one-sample-ahead conditional variance and clean signal estimation, i.e.,

$$\begin{aligned}\hat{x}_t &= E\left\{x_t|y_1^t\right\} \\ &= E\left\{x_t|\hat{x}_{t-p}^{t-1},\hat{\sigma}_{t|t-1}^2,y_t\right\}\end{aligned} \tag{6.9}$$

$$\begin{aligned}\hat{\sigma}_{t|t-1}^2 &= E\left\{\sigma_{t|t-1}^2|y_1^{t-1}\right\} \\ &= E\left\{\sigma_{t|t-1}^2|\hat{x}_{t-p-1}^{t-2},\hat{\sigma}_{t-1|t-2}^2,y_{t-1}\right\}.\end{aligned} \tag{6.10}$$

To estimate the clean signal, note that

$$\begin{aligned}\left(x_t|y_1^{t-1}\right) &\sim \left(x_t|\hat{x}_{t-p}^{t-1},\hat{\sigma}_{t|t-1}^2\right) \\ &\sim \mathcal{N}\left(\sum_{i=1}^{p}\alpha_i\hat{x}_{t-i},\hat{\sigma}_{t|t-1}^2\right)\end{aligned} \tag{6.11}$$

and

$$\left(n_t|y_1^{t-1}\right) \sim \mathcal{N}\left(0,\sigma_t^2\right). \tag{6.12}$$

Using (6.4), (6.9), (6.11), (6.12) and the well-known estimate of Gaussian signals in Gaussian noise [66] we get

$$\begin{aligned}\hat{x}_t &= E\left\{x_t|\hat{x}_{t-p}^{t-1},\hat{\sigma}_{t|t-1}^2,y_t\right\} \\ &= \sum_{i=1}^{p}\alpha_i\hat{x}_{t-i}+\frac{\hat{\sigma}_{t|t-1}^2}{\hat{\sigma}_{t|t-1}^2+\sigma_t^2}\left(y_t-\sum_{i=1}^{p}\alpha_i\hat{x}_{t-i}\right).\end{aligned} \tag{6.13}$$

Substituting (6.3) into (6.10) we get

$$\begin{aligned}\hat{\sigma}_{t|t-1}^2 &= E\left\{\sigma_{t|t-1}^2|\hat{x}_{t-p-1}^{t-2},\hat{\sigma}_{t-1|t-2}^2,y_{t-1}\right\} \\ &= E\left\{\beta_0+\beta_1\varepsilon_{t-1}^2+\beta_2\sigma_{t-1|t-2}^2|\hat{x}_{t-p-1}^{t-2},\hat{\sigma}_{t-1|t-2}^2,y_{t-1}\right\} \\ &= \beta_0+\beta_1\underbrace{E\left\{\varepsilon_{t-1}^2|\hat{x}_{t-p-1}^{t-2},\hat{\sigma}_{t-1|t-2}^2,y_{t-1}\right\}}_{u_{t-1}}+\beta_2\hat{\sigma}_{t-1|t-2}^2.\end{aligned} \tag{6.14}$$

The last step to complete the formulation of the MMSE estimation of the clean signal is to compute u_{t-1} as follows

$$\begin{aligned} u_{t-1} &= E\left\{\varepsilon_{t-1}^2 | \hat{x}_{t-p-1}^{t-2}, \hat{\sigma}_{t-2|t-2}^2, y_{t-2}\right\} \\ &= E\left\{\left(x_{t-1} - \sum_{i=1}^{p} \alpha_i x_{t-1-i}\right)^2 \Big| \hat{x}_{t-1-p}^{t-2}, \hat{\sigma}_{t-1|t-2}^2, y_{t-2}\right\} \\ &= E\left\{\left(x_{t-1} - \sum_{i=1}^{p} \alpha_i \hat{x}_{t-1-i}\right)^2 \Big| \hat{\sigma}_{t-1|t-2}^2, y_{t-1}\right\} \end{aligned} \tag{6.15}$$

The last expectation in (6.15) can be easily computed using posterior pdf of the Bayesian general linear model [66] [equations (10.28)-(10.29)] as follows

$$\begin{aligned} u_{t-1} &= E\left\{\left(x_{t-1} - \sum_{i=1}^{p} \alpha_i \hat{x}_{t-1-i}\right)^2 \Big| \hat{\sigma}_{t-1|t-2}^2, y_{t-1}\right\} \\ &= \operatorname{var}\left\{x_{t-1} \Big| \hat{\sigma}_{t-1|t-2}^2, y_{t-1}\right\} + \left(E\left\{x_{t-1} \Big| \hat{\sigma}_{t-1|t-2}^2, y_{t-1}\right\} - \sum_{i=1}^{p} \alpha_i \hat{x}_{t-1-i}\right)^2 \\ &= \left(\hat{\sigma}_{t-1|t-2}^2 - \frac{\hat{\sigma}_{t-1|t-2}^4}{\hat{\sigma}_{t-1|t-2}^2 + \sigma_{t-1}^2}\right) + \left(\frac{\hat{\sigma}_{t-1|t-2}^4}{\hat{\sigma}_{t-1|t-2}^2 + \sigma_{t-1}^2}\left(y_{t-1} - \sum_{i=1}^{p} \alpha_i \hat{x}_{t-1-i}\right)\right)^2 \\ &= \frac{\hat{\sigma}_{t-1|t-2}^2 \sigma_{t-1}^2}{\hat{\sigma}_{t-1|t-2}^2 + \sigma_{t-1}^2} + \left(\frac{\hat{\sigma}_{t-1|t-2}^2}{\hat{\sigma}_{t-1|t-2}^2 + \sigma_{t-1}^2}\right)^2 \left(y_{t-1} - \sum_{i=1}^{p} \alpha_i \hat{x}_{t-1-i}\right)^2. \end{aligned} \tag{6.16}$$

So far we have assumed that the parameters are known and we have found the MMSE estimate of the clean signal and the one-sample-ahead conditional variance. In real applications, this assumption is often not true, and we have to estimate the parameters of the model from the noisy observations. The ML estimate of the parameters is obtained by solving the following nonlinear optimization problem

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{ML} &= \max_{\boldsymbol{\theta}} \log\left(f\left(y_1^N; \boldsymbol{\theta}\right)\right) \\ \log\left(f\left(y_1^N; \boldsymbol{\theta}\right)\right) &= \sum_{t=2}^{N} \log\left(f\left(y_t | y_1^{t-1}; \boldsymbol{\theta}\right)\right) + \log\left(f(y_1; \boldsymbol{\theta})\right). \end{aligned} \tag{6.17}$$

where

$$\log\left(f\left(y_1^N; \boldsymbol{\theta}\right)\right) = \sum_{t=2}^{N} \log\left(f\left(y_t | y_1^{t-1}; \boldsymbol{\theta}\right)\right) + \log\left(f(y_1; \boldsymbol{\theta})\right) \tag{6.18}$$

and $f\left(y_1^N; \boldsymbol{\theta}\right)$ and $\hat{\boldsymbol{\theta}}_{ML}$ are, respectively, the likelihood function given the parameters and the maximum likelihood estimate of the parameters. The maximization problem described in (6.17)

has no closed form solution, so it must be solved by numerical methods such as the steepest decent [76]. The main drawback of the steepest decent method is its high computational load making it inapplicable in real-time applications such as speech enhancement. In order to overcome this problem, a procedure like the one presented in [84] can be used to estimate the parameters together with the clean signal adaptively. The idea behind this method is to update the gradient of the likelihood function each time we receive new data. Next, the algorithm uses this estimate of the gradient vector to update the estimate of the parameter vector in the same way as the steepest decent method.

Under suitable regularity conditions described in [117] it can be shown that the average log-likelihood converges to the following limit

$$l(\boldsymbol{\theta}) = \lim_{k\to\infty} \frac{1}{k+1} \sum_{t=1}^{k} \log\left[f\left(\mathbf{y}^t|\mathbf{y}_0^{t-1};\boldsymbol{\theta}\right)\right]. \tag{6.19}$$

It can also be shown that $l(\boldsymbol{\theta})$ admits $\boldsymbol{\theta}^*$ as a global maximum where $\boldsymbol{\theta}^*$ is the global maximum of the log-likelihood function in (6.17) [75]. To maximize $l(\boldsymbol{\theta})$, one can use a Stochastic Approximation (SA) algorithm to update the parameter estimate at time t using the following recursion formula

$$\hat{\boldsymbol{\theta}}_t = \hat{\boldsymbol{\theta}}_{t-1} + \mu_t \nabla \log\left[f\left(\mathbf{y}_t|\mathbf{y}_0^{t-1};\hat{\boldsymbol{\theta}}\right)\right] \tag{6.20}$$

where $\hat{\boldsymbol{\theta}}_{t-1}$ is the parameter estimate at time $t-1$, and $\mathbf{g}_t = \nabla \log\left[f\left(\mathbf{y}_t|\mathbf{y}_0^{t-1};\hat{\boldsymbol{\theta}}\right)\right]$ is computed in the Appendix. This method is called recursive maximum likelihood (RML), and the algorithm for this method is summarized in Table 6.1. In this table, the step size, μ_t is a positive non-increasing sequence, such that

$$\lim_{T\to\infty} \sum_{t=1}^{T} \mu_t = \infty \tag{6.21}$$

$$\sum_{t=1}^{\infty} \mu_t^2 < \infty. \tag{6.22}$$

It can be shown that $\boldsymbol{\theta}_t$ will converge to the set of (global or local) maxima of $\log f\left(y_1^N;\boldsymbol{\theta}\right)$ [75]. It is worth mentioning that the RML method can also be used for adaptive parameter estimation (varying parameters). In this case, the choice of μ_t will be a trade-off between tracking capability (large μ_t) and low estimation noise around the parameter (small μ_t).

Table 6.1: Recursive ML Algorithm for Estimating the Parameters of a Noisy AR-GARCH Process

Initialization:

Let $\hat{\boldsymbol{\theta}}_0 = \mathbf{0}$.

Let $u_t = \hat{\sigma}^2_{t-1|t-2} = \hat{x}_t = 0$ for $t = 1, 2, ..., p$.

for $t = p+1$ to N

Compute $\hat{\sigma}^2_{t|t-1} = \beta_0 + \beta_1 u_{t-1} + \beta_2 \hat{\sigma}^2_{t-1|t-2}$.

Compute $\mathbf{g}_t$ using (6.41).

$\hat{\boldsymbol{\theta}}_t = \hat{\boldsymbol{\theta}}_{t-1} + \mu_t \mathbf{g}_t$.

Compute

$$u_t = \frac{\hat{\sigma}^2_{t|t-1}\sigma^2_t}{\hat{\sigma}^2_{t|t-1}+\sigma^2_t} + \left(\frac{\hat{\sigma}^2_{t|t-1}}{\hat{\sigma}^2_{t|t-1}+\sigma^2_t}\right)^2 \left(y_t - \sum_{i=1}^p \alpha_i \hat{x}_{t-i}\right)^2.$$

$$\hat{x}_t = \sum_{i=1}^p \alpha_i \hat{x}_{t-i} + \frac{\hat{\sigma}^2_{t|t-1}}{\hat{\sigma}^2_{t|t-1}+\sigma^2_t}\left(y_t - \sum_{i=1}^p \alpha_i \hat{x}_{t-i}\right).$$

end (for)

6.2.1 Considering the Stationarity Conditions

Often in signal processing applications, it is needed that the estimated model be stationary. Hence, we must maximize the log-likelihood function under stationarity conditions. The AR-GARCH process is strictly stationary with finite second order moment if and only if the following conditions hold [74]:

i) $\beta_0 > 0$ and $\beta_i \geq 0 \; \forall \; 1 \leq i \leq 2$

ii) $\beta_1 + \beta_2 < 1$

iii) All the roots of $1 - \sum_{i=1}^p \alpha_i z^{-i}$ must be inside the unit circle.

It is obvious that the third condition is highly nonlinear and cannot be easily applied to nonlinear optimization [76]. To overcome this problem, we use an alternative condition for the stationarity of AR processes that the roots of $1 - \sum_{i=1}^p \alpha_i z^{-i}$ are inside the unit circle (so that the AR process is stationary) if, and only if, $|\gamma_i| \leq 1 \; \forall \; 1 \leq i \leq p$ where γ_i's are the reflection coefficients of the AR model [115]. The reflection coefficients of the AR model can be computed uniquely from the AR coefficients recursively using the Levinson-Durbin recursion algorithm as follows

[115]

$$\begin{aligned}
\alpha_{i,k} &= \alpha_{i,k-1} + \gamma_k \alpha_{k-i,k-1}, \quad 1 \leq i < k \\
\alpha_{k,k} &= \gamma_k, \quad k = 1, ..., p \\
\alpha_i &= -\alpha_{i,p}, \quad i = 1, ..., p.
\end{aligned} \tag{6.23}$$

Therefore, it can be concluded that the AR-GARCH process is strictly stationary with finite second order moment if, and only if, i, ii, and the following conditions hold:

iv) $-1 < \gamma_i < 1$,$\forall\ 1 \leq i \leq p$.

Obviously, three inequality constraints (i.e., i, ii and iv) are linear, and the ML estimate of the parameters can be found by solving the following optimization problem

$$\begin{aligned}
\hat{\phi} &= \arg \max_{\phi} \left\{ \log \left(f \left(y_1^N ; \phi \right) \right) \right\} \\
&\quad \text{s.t.} \\
&\quad \beta_0 > 0 \ \text{ and } \ \beta_i \geq 0 \ \ \forall 1 \leq i \leq 2 \\
&\quad \beta_1 + \beta_2 < 1 \\
&\quad 1 < \gamma_i < -1 \ \ \forall 1 \leq i \leq p
\end{aligned} \tag{6.24}$$

where $\phi = [\beta_0, \beta_1, \beta_2, \gamma_1, \gamma_2, ..., \gamma_p]^T$ is the vector of parameters consisting of the parameters of the GARCH part and reflection coefficients of the AR part.

It is clear that the maximization problem described in (6.24) has no closed form solution and must be solved using numerical methods such as the gradient projection method [76]. As previously stated, this optimization problem has a high computational load; therefore, we use the RML method in order to decrease computational complexity. In order to use these numerical methods, we must compute the gradient of the log-likelihood function of the current observation conditioned on past observations with respect to the recently defined vector of parameters (ϕ). The first three elements of the gradient vector are computed in the Appendix (equations (6.37) through (6.41)), and the other elements can be computed using the following equations

$$\begin{aligned}
g_{t,i+3} &= \frac{\partial}{\partial \gamma_i} \left(\log \left(f \left(y_t | y_0^{t-1} ; \phi \right) \right) \right) \\
&= \sum_{k=1}^{p} \frac{\partial}{\partial \alpha_k} \left(\log \left(f \left(y_t | y_0^{t-1} ; \phi \right) \right) \right) \frac{\partial \alpha_k}{\partial \gamma_i} \\
&\qquad i = 1, 2, ..., p
\end{aligned} \tag{6.25}$$

where the partial derivative of the log-likelihood function with respect to the AR parameters (i.e., $\frac{\partial}{\partial \alpha_k}\left(\log\left(f\left(y_t|y_0^{t-1};\phi\right)\right)\right)$) is computed in the Appendix (equation (6.41)), and the partial derivative of α_k with respect to γ_i (i.e. $\frac{\partial \alpha_k}{\gamma_i}$) can be obtained recursively from the following equations [126]:

$$\begin{aligned}
\frac{\partial \alpha_{k,i}}{\partial \gamma_i} &= \begin{cases} \alpha_{i-k,i-1} & \text{if } 1 \leq k < i \\ 1 & \text{if } k = i \end{cases} \\
\frac{\partial \alpha_{k,j}}{\partial \gamma_i} &= \begin{cases} \frac{\partial \alpha_{k,j-1}}{\partial \gamma_i} + \gamma_j \frac{\partial \alpha_{j-k,j-1}}{\partial \gamma_i} & \text{if } 1 \leq k < j \\ 0 & \text{if } k = i \text{ and } i < k \leqslant p. \end{cases} \\
\frac{\partial \alpha_k}{\partial \gamma_i} &= -\frac{\partial \alpha_{k,p}}{\partial \gamma_i}
\end{aligned} \tag{6.26}$$

In the next section, we use the stationarity conditions along with the RML parameter estimation method in order to propose a new VAD.

6.3 VAD based on AR-GARCH modeling of speech signals in time domain

In this section, we introduce our VAD, which is based on an AR-GARCH model. We assume that the speech signal in the time domain can be modeled by an AR-GARCH process. Let y_t be the noisy samples of the speech signal in the time domain divided by the noise variance, σ_t^2. The noise variance, σ_t^2, is estimated by the IMCRA method and equations (6.6)-(6.7). This makes the proposed VAD a constant false alarm rate (CFAR) VAD, without affecting the overall performance of the algorithm. Suppose that we have a time frame and we want to decide whether it consists of speech or not. Let m be the frame index, K be the number of samples in each frame, and $0 \leq \delta \leq 1$ be the overlap factor between adjacent frames. The likelihood ratio (LR) for the t-th sample, given the observations up to $t-1$, can be written as follows

$$\lambda_t = \frac{f\left(y_t|y_1^{t-1};\boldsymbol{\theta},\boldsymbol{H}_1\right)}{f\left(y_t|y_1^{t-1};\boldsymbol{\theta},\boldsymbol{H}_0\right)} \tag{6.27}$$

where $\boldsymbol{H}_1$ and $\boldsymbol{H}_0$ are the speech presence and absence hypotheses, respectively. The decision rule for the m-th time frame is established from the geometric mean of the likelihood ratios for

the individual time sample [111], which is given by

$$\begin{aligned}\log\Lambda_m &= \frac{1}{K+2K\delta}\sum_{t=t_{1,m}}^{t_{2,m}}\log\lambda_t \\ &= \frac{1}{K+2K\delta}\sum_{t=t_{1,m}}^{t_{2,m}}\log\frac{f\left(y_t|y_1^{t-1};\boldsymbol{\theta},\boldsymbol{H}_1\right)}{f\left(y_t|y_1^{t-1};\boldsymbol{\theta},\boldsymbol{H}_0\right)} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} T_h\end{aligned} \tag{6.28}$$

where $t_{1,m} = (m-1-\delta)K+1$ and $t_{2,m} = (m+\delta)K$. Suppose that the estimate of the model parameters ($\hat{\boldsymbol{\theta}}_t$), the one-sample-ahead conditional variance of the speech signal ($\hat{\sigma}^2_{t|t-1}$), and the estimate of the clean signal ($\hat{x}_t$) are obtained using the RML procedure. The numerators of the terms in the right side of (6.27) can be computed using (6.37). Assuming that the noise is white Gaussian with known variance, the denominators of the terms in the right side of (6.27) can be computed by

$$f\left(y_t|y_1^{t-1};\boldsymbol{\theta},\boldsymbol{H}_0\right) = f\left(y_t;\boldsymbol{\theta},\boldsymbol{H}_0\right) = \frac{1}{\sqrt{2\pi\sigma_t^2}}\exp\left(\frac{-y_t^2}{2\sigma_t^2}\right). \tag{6.29}$$

Another issue that must be taken into account is the correlation between consecutive samples of the speech signal. The sequence of speech activity states can be modeled as a first-order Markov process which may help to prevent clipping of weak speech [111]. Based on the total probability theorem and Bayes' rule, we can easily derive the soft VAD rule as follows [51]

$$P_{m|m} = \frac{\Lambda_m P_{m|m-1}}{\Lambda_m P_{m|m-1} + (1-P_{m|m-1})} \tag{6.30}$$

where m is the index of a time frame consisting of k samples, Λ_m is the likelihood ratio of the m-th frame, which can be computed by (6.27), and $P_{m|m}$ and $P_{m|m-1}$ are the soft decision rule with and without using the information provided by the m-th frame data, respectively. The relationship between $P_{m|m}$ and $P_{m|m-1}$ is similar to the relationship between the current state and the previous state in the Markov model and is given by [51]

$$P_{m+1|m} = h_{0,1}(1-P_{m|m}) + h_{1,1}P_{m|m} \tag{6.31}$$

where $0 < h_{0,1} < 1$ and $0 < h_{1,1} < 1$ denote the probabilities of speech activity in the current frame when there is silence or speech in the previous frame, respectively. The issue of selecting the values of $h_{0,1}$ and $h_{1,1}$ is a trade-off between the probability of false alarm ($\mathrm{P_{fa}}$) and the

probability of missed detection ($\mathrm{P_{miss}}$). This means that a smaller value of $h_{0,1}$ results in less noise detected as signal (smaller probability of false alarm) at the expense of more front-end speech being detected as noise (greater $\mathrm{P_{miss}}$) and that a greater value of $h_{1,1}$ cause less middle speech clipping at the expense of more front-end speech being detected as noise.

Our final voice activity detector is obtained by comparing $P_{m|m}$ to a predetermined threshold, i.e.,

$$H(m) = \begin{cases} H_1, & \text{if } P_{m|m} \geq T_h \\ H_0, & \text{otherwise} \end{cases} \tag{6.32}$$

where T_h determines the trade-off between the probability of false alarm ($\mathrm{P_{fa}}$) and the probability of detection ($\mathrm{P_d}$). The proposed VAD algorithm is summarized in Table 6.2. In this table, ϕ^i is the i-th element of the vector ϕ. As stated in Table 6.2, the VAD algorithm first utilizes the IMCRA method to estimate the noise variance. A new observation sequence (y_t) is then defined by dividing the the noisy speech sequence by the estimated noise variance obtained by the IMCRA method. Then, the RML method is used to estimate the parameters of the AR-GARCH model together with the hidden states and the one-sample-ahead variance of the speech signal. Finally the likelihood ratio is computed using the estimated one-sample-ahead variance of the speech signal and is compared to a threshold.

6.4 Performance Evaluation

In this section, we evaluate the performance of the proposed parameter estimation method and VAD under several conditions. In the first experiment, we compare the performance of the constrained ML parameter estimation method (i.e., optimization problem in (6.24) solved using gradient the projection method [76]) with those of two different ML methods. The first method, denoted by MLClean, employs clean data (unavailable in practical situations) for estimating the parameter. The second method, denoted by MLNoisy, utilizes the noisy data for estimating the parameter but falsely handles the data as clean. In this experiment, we use ten different AR(2)-GARCH(1,1), processes corrupted with zero-mean complex Gaussian white noise with two different SNR levels. The number of available data (N) is set to 1024. The process v_t is a zero-mean Gaussian white process with unity variance. The reflection coefficients of the AR part are chosen randomly and uniformly from the interval $(-1,1)$, and the parameters of the

Table 6.2: The Proposed VAD Algorithm Using AR-GARCH Modeling

Initialization:

Use the IMCRA method presented in section 8.2, to estimate the variance of the noise $\hat{\sigma}_t^2$

Let y_t be the noisy speech signal divided by $\hat{\sigma}_t$.

Let $\sigma_t^2 = 1$, $\phi_0 = \mathbf{0}$ and $u_t = \hat{\sigma}_{t-1|t-2}^2 = \hat{x}_t = 0$ for $t = 1, 2, ..., p+1$.

for $t = p+2$ to N.

Compute the AR parameters from the reflection coefficients using (6.23).

Compute $\hat{\sigma}_{t|t-1}^2 = \beta_0 + \beta_1 u_{t-1} + \beta_2 \hat{\sigma}_{t-1|t-2}^2$.

Compute $\mathbf{g}_t$ using (6.41), (6.25) and (6.26).

$\hat{\phi}_t = \hat{\phi}_{t-1} + \mu_t \mathbf{g}_t$.

If $\hat{\phi}_t^i \leq 0; 1 \leq i \leq 3$ then $\hat{\phi}_t^i = 0$.

If $\hat{\phi}_t^i \geq 1; 1 \leq i \leq 3$ then $\hat{\phi}_t^i = 1$.

If $\hat{\phi}_t^i \leq -1; 4 \leq i \leq 3+p$ then $\hat{\phi}_t^i = -1$.

If $\hat{\phi}_t^i \geq 1; 4 \leq i \leq 3+p$ then $\hat{\phi}_t^i = 1$.

Compute

$u_t = \frac{\hat{\sigma}_{t|t-1}^2 \sigma_t^2}{\hat{\sigma}_{t|t-1}^2 + \sigma_t^2} + \left(\frac{\hat{\sigma}_{t|t-1}^2}{\hat{\sigma}_{t|t-1}^2 + \sigma_t^2} \right)^2 \left(y_t - \sum_{i=1}^p \alpha_i \hat{x}_{t-i} \right)^2$.

$\hat{x}_t = \sum_{i=1}^p \alpha_i \hat{x}_{t-i} + \frac{\hat{\sigma}_{t|t-1}^2}{\hat{\sigma}_{t|t-1}^2 + \sigma_t^2} \left(y_t - \sum_{i=1}^p \alpha_i \hat{x}_{t-i} \right)$.

end (for)

Select K, $h_{0,1}$ and $h_{1,1}$ and let $P_{1|0} = \frac{1}{2}$.

for $m = 1$ to $\left[\frac{N}{K}\right] - 1$.

Compute

$\Lambda_m = \exp\left(\frac{1}{K+2K\delta} \sum_{t=t_{1,m}}^{t_{2,m}} \log \frac{f\left(y_t | y_1^{t-1}; \boldsymbol{\theta}, \boldsymbol{H}_1\right)}{f\left(y_t | y_1^{t-1}; \boldsymbol{\theta}, \boldsymbol{H}_0\right)} \right)$

$P_{m|m} = \frac{\Lambda_m P_{m|m-1}}{\Lambda_m P_{m|m-1} + (1 - P_{m|m-1})}$.

Update $P_{m+1|m} = h_{0,1}(1 - P_{m|m}) + h_{1,1} P_{m|m}$.

Decide whether the m-th frame contains speech or not by comparing $P_{m|m}$ with a threshold.

end (for)

GARCH$(1,1)$ are chosen randomly and uniformly from the interval $(0,1)$ such that the processes will be stationary. In this experiment, the additive corrupting noise variance is assumed

to be known. For evaluating the performance of the proposed method we use mean square errors (MSE) in estimation of the parameters (which were estimated using 1000 realizations) and MSE for the estimate of $\sigma^2_{t|t-1}$, which are given by

$$\text{MSE} = \frac{1}{10}\sum_{j=1}^{10}\frac{1}{10^3}\sum_{i=1}^{1000}\frac{\left(\hat{\theta}_{k,(i,j)}-\theta_{k,j}\right)^2}{\left(\theta_{k,j}\right)^2} \tag{6.33}$$

$$E_{\sigma^2_t} = \frac{1}{10^4}\sum_{j=1}^{10}\sum_{i=1}^{1000}\frac{1}{N-1}\sum_{t=2}^{N}\left(\sigma^2_{t|t-1,(i,j)}-\hat{\sigma}^2_{t|t-1,(i,j)}\right)^2 \tag{6.34}$$

where i is the iteration index, j is the index for different parameters set, $\theta_{k,j}$ is the k-th element of the vector of parameters of the j-th process, $\hat{\theta}_{k,(i,j)}$ is the estimate of $\theta_{k,j}$ in i-th iteration, and $\hat{\sigma}^2_{t|t-1,(i,j)}$ is computed as follows

$$\hat{\sigma}^2_{t|t-1,(i,j)} = \hat{\theta}_{0,(i,j)} + \hat{\theta}_{1,(i,j)}\varepsilon^2_{t-1} + \hat{\theta}_{2,(i,j)}\hat{\sigma}^2_{t-1|t-2,(i,j)}. \tag{6.35}$$

The results are given in Table 6.3. It is obvious that the proposed method yields better performance over the MLNoisy method often used in real world problems. An important point to be emphasized here is that all of these methods (i.e. constrained ML, MLNoisy, MlNoisy) have the similar computational loads. This makes these methods unsuitable for real-time applications. We give these results for the sake of comparison.

Table 6.3: MSE in Parameter Estimation and MSE in Estimation of $\sigma^2_{t|t-1}$ for Different SNR Levels

Method:	MLClean	MLNoisy		Proposed	
SNR	——	5dB	10dB	5dB	10dB
β_0	0.1791	26.0113	3.4975	3.8168	1.2206
β_1	0.1023	0.7089	0.4879	0.4192	0.2400
β_2	0.1521	0.6499	0.4412	0.7246	0.5376
α_1	0.0048	0.1102	0.0203	0.0082	0.0065
α_2	0.0002	0.0596	0.0086	0.0003	0.0002
E^2_σ	0.0086	2.2257	0.3973	0.2009	0.0554

In the second experiment, we compare the performance of the proposed VAD with those of four commonly used VADs (i.e., Sohn et al. [111], Ramirez et al. [96], Chang et al. [21] and Shin et al. [108]). The speech signal used in our evaluation is taken from the TIMIT database

[50]. We use a speech signal of 12s long consisting of different English sentences from both male and female speakers sampled at 16 kHz. The speech signal is corrupted by a computer generated white Gaussian noise, and the overall SNR is set to 5 dB. The parameters of the proposed VAD are given in Table 6.4. These parameters are chosen experimentally to achieve the highest performance.

Table 6.4: Parameter Values Used for the Proposed VAD

K	μ_t	$h_{0,1}$	$h_{1,1}$	p	δ
256	$\frac{0.1}{\lVert \mathbf{g}_t \rVert}$	0.80	0.90	1	.90

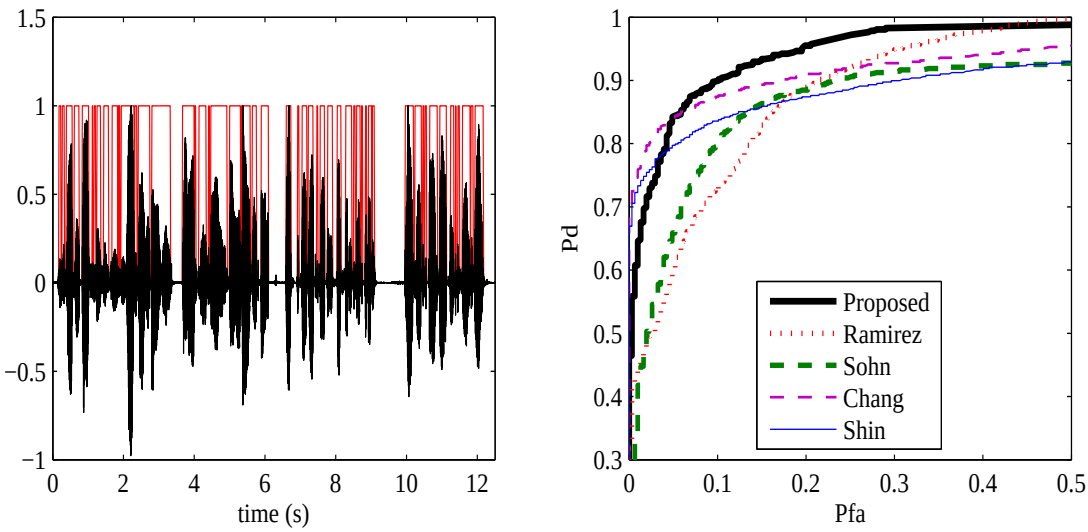
The results of the simulations are depicted in Fig. 6.3 (a) and Fig. 6.3 (b). The clean speech signal together with a hand-marked VAD is depicted in Fig. 1. The hand-marked VAD is obtained as follows. In the m-th time frame we assume that the speech is present whenever $10\log_{10} S_m > \max_m\{10\log_{10} S_m\} - 22$ and the speech is absent whenever $10\log_{10} S_m < \max_m\{10\log_{10} S_m\} - 23$ where

$$S_m = \frac{1}{k} \sum_{t=(m-1)k+1}^{mk} x_t^2.$$

No decision is made in other frames. We chose these hand-marked VAD thresholds experimentally by two qualitative criteria. The speech presence threshold (-22 in this case) is chosen such that the listeners do not sense any essential difference between the original clean signal and the speech signal passed through the hand-marked VAD. The speech absence threshold (-23 in this case) is chosen such that the listeners do not sense any speech signal component in that part of speech which is labeled as silence.

In this experiment, the unknown variance of the additive noise is estimated using IMCRA for all of the reference methods. Fig. 6.3 (b) shows the receiver operating characteristic (ROC) of the four mentioned methods together with that of the proposed method. The ROCs show the probability of detection ($\mathrm{P_d}$) versus the probability of false alarms ($\mathrm{P_d}$) where $\mathrm{P_d}$ is the probability that a signal frame is classified as H_1 and $\mathrm{P_{fa}}$ is the probability that a noise frame is classified as H_1. These ROCs are obtained by 100 repetitions of the experiment with the same speech signal and different white Gaussian noise. From these curves, it is obvious that our method outperforms the VADs proposed in [111] and [96] for all values of probability of false

alarm. Fig. 6.3 (b) also shows that although the performance of the method proposed in [21] and [108] is a little bit better than that of the proposed method for low probability of false alarm, the performance of the proposed method is much better than that of the method proposed in [21] and [108] for high probability of false alarm. In Table 6.5, we also provide the probability of detection for different values of probability of false alarm and different SNR levels for the above mentioned methods. In Table 6.5, the best performance in each column (i.e., different SNRs) is indicated by bold numbers. As can be seen from the data in this table, the performance of the proposed method is higher than that of other methods except for low probability of false alarm in high SNRs, as mentioned before. As can be seen from the first column of Table 6.5, the proposed VAD also outperforms the reference VADs even in low probability of false alarm in low SNRs.



(a) (b)

Figure 6.3: (a) Clean speech signal together with hand-marked VAD. (b) Comparison of ROC curves of the proposed method with those of reference methods for Gaussian noise and SNR=5dB.

In the third experiment, we investigate the effect of other kinds of noise on the performance of the proposed VAD. In this simulation the parameters of the proposed VAD are chosen from Table 6.2. Two noise types (factory and babble) are employed in this experiment. These two noise types are generally difficult to deal with. For all the reference methods, the IMCRA method is used for estimating the variance of the noise in the STFT domain. Figs. 6.4 (a) and 6.4 (b) show the receiver operating characteristic (ROC) of the four mentioned methods together

Table 6.5: Probability of Detection Versus Probability of False Alarm for Different SNR Levels and Different Methods (White Noise)

	$P_{fa} = 5\%$			$P_{fa} = 10\%$			$P_{fa} = 20\%$			$P_{fa} = 40\%$		
SNR	0db	5db	10db	0db	5db	10db	0db	5db	10db	0db	5db	10db
$P_d$for Proposed Method	**80.49**	**84.12**	85.93	**84.43**	**87.10**	89.13	**87.31**	**89.66**	90.72	**87.63**	**91.47**	**92.96**
$P_d$for Ramirez Method	51.60	57.78	64.18	61.09	64.61	70.04	69.51	73.13	76.97	73.99	77.83	81.56
$P_d$for Sohn Method	56.93	66.52	69.51	64.29	72.92	75.69	69.40	77.51	80.28	71.43	79.32	81.77
$P_d$for Chang Method	67.59	82.52	**88.70**	70.36	84.22	**90.19**	73.88	86.25	**91.26**	77.83	88.27	92.43
$P_d$for Shin Method	63.33	78.68	86.99	65.99	80.92	88.27	69.62	82.94	89.45	73.88	84.97	90.41

with that of the proposed method for factory and babble noise, respectively. We also provide in Tables 6.6 and 6.7 detection probability for different values of probability of false alarm and for different SNR levels. In these tables, the best performance in each column (i.e., different SNRs) is indicated by bold numbers. The figures and tables are obtained using the same method explained in the second experiment. The advantage of the proposed method over the reference methods is obvious from these results.

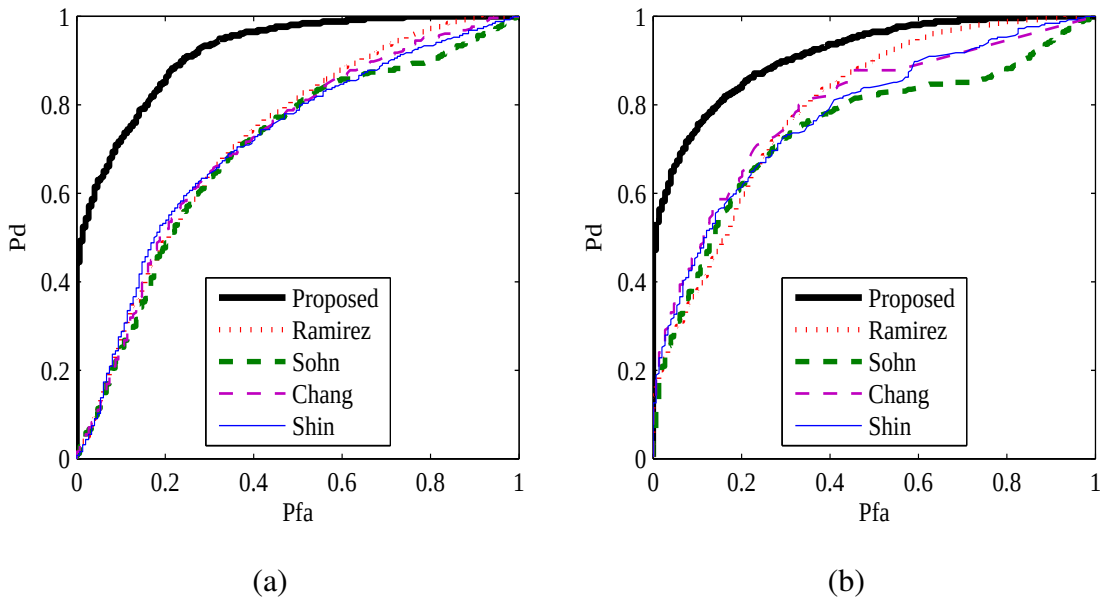


Figure 6.4: (a) Comparison of ROC curves of the proposed method with those of reference methods for factory noise and SNR=5dB. (b) Comparison of ROC curves of the proposed method with those of reference methods for babble noise and SNR=5dB.

In the forth experiment, the effect of the order of the autoregressive part on the performance of the VAD is investigated. The ROC curves for different values of p are depicted in Fig. 6.5

Table 6.6: Probability of Detection Versus Probability of False Alarm for Different SNR Levels and Different Methods (Factory Noise)

	$P_{fa} = 5\%$			$P_{fa} = 10\%$			$P_{fa} = 20\%$			$P_{fa} = 40\%$		
SNR	0db	5db	10db	0db	5db	10db	0db	5db	10db	0db	5db	10db
$P_d$for Proposed Method	**54.69**	**73.03**	**81.34**	**59.06**	**76.87**	**84.86**	**64.39**	**80.28**	**87.31**	**70.04**	**82.94**	**88.59**
$P_d$for Ramirez Method	6.82	14.29	23.24	12.05	21.32	30.92	20.90	31.98	40.09	35.93	45.31	51.81
$P_d$for Sohn Method	7.68	13.86	22.81	12.47	21.00	30.92	21.00	31.77	40.94	34.75	43.92	52.24
$P_d$for Chang Method	7.25	14.50	26.23	12.47	22.17	34.33	22.39	33.26	43.39	35.82	44.46	52.67
$P_d$for Shin Method	7.46	14.71	26.87	14.61	23.56	35.18	24.63	33.48	44.67	36.89	44.78	53.52

Table 6.7: Probability of Detection Versus Probability of False Alarm for Different SNR Levels and Different Methods (Babble Noise)

	$P_{fa} = 5\%$			$P_{fa} = 10\%$			$P_{fa} = 20\%$			$P_{fa} = 40\%$		
SNR	0db	5db	10db	0db	5db	10db	0db	5db	10db	0db	5db	10db
$P_d$for Proposed Method	**53.20**	**73.13**	**80.81**	**57.89**	**75.59**	**84.22**	**63.22**	**79.32**	**86.67**	**69.08**	**82.73**	**88.38**
$P_d$for Ramirez Method	27.29	35.61	40.72	33.90	41.90	46.16	43.07	50.11	53.30	53.30	60.98	65.14
$P_d$for Sohn Method	27.08	34.97	38.38	33.48	42.22	45.74	42.43	51.17	55.54	50.85	58.53	61.94
$P_d$for Chang Method	28.78	39.77	44.14	34.65	45.95	52.45	43.39	55.76	60.23	51.60	61.09	62.47
$P_d$for Shin Method	27.93	39.34	44.56	34.01	45.42	51.28	41.26	52.45	59.49	49.15	58.74	62.15

(a). These curves are obtained by the same setup explained in the second experiment (i.e., white Gaussian noise and SNR=5 dB and the same set of parameters for the proposed VAD) but a different speech signal is used. The speech signals were again chosen from the TIMIT database, half from male and half from female speakers with total length of 20s. As can be seen from this figure, the performances of the different AR-GARCH models are approximately the same for $1 \leq p \leq 4$ and get worse for a higher order. This is because increasing the order (increasing the number of parameters) makes the estimation of the parameters more difficult, and larger error in estimation of the parameters impairs the performance of the proposed VAD. Thus, we chose the first order in other simulations because this model is computationally simpler than higher order models.

In the final experiment, we evaluate the performance of the proposed VAD in presence of an additive sinusoidal component. A 20s long speech signal is chosen from the TIMIT database and corrupted by a sinusoidal signal of 60 Hz frequency. The SNR is set to 5 dB. The ROC curves for the reference methods are depicted in Fig. 6.5 (b). For all reference methods, the

IMCRA method is used for estimating the variance of the additive noise. We also provide in Table 6.8 detection probability for different values of probability of false alarm and for different SNR levels. In this table, the best performance in each column (i.e., different SNRs) is indicated by bold numbers. It is apparent that the performance of the proposed VAD is substantially higher than that of the reference VADs.

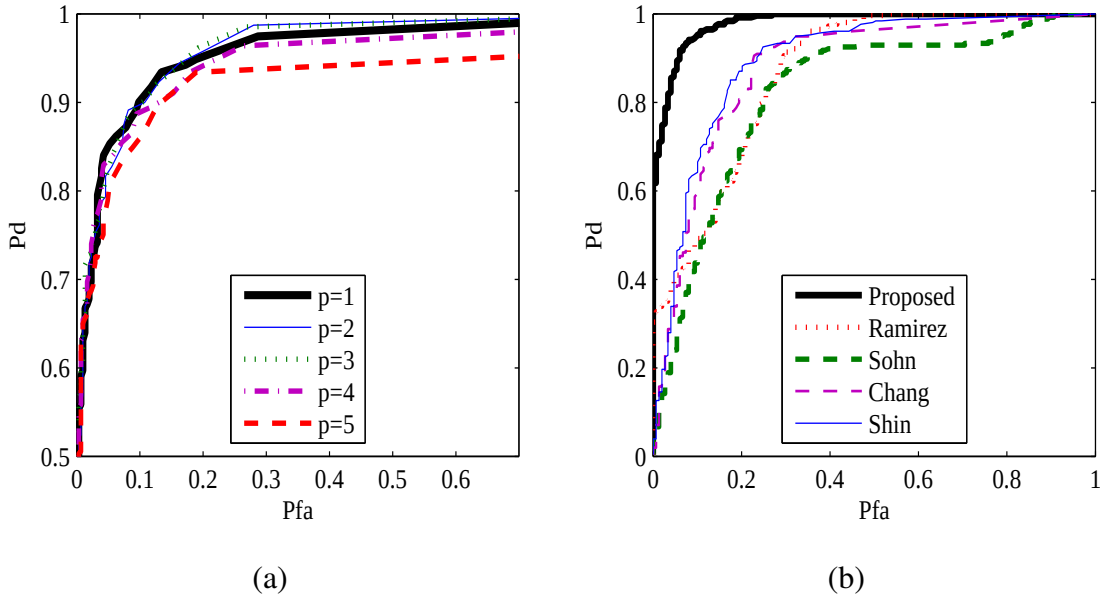


Figure 6.5: (a) ROC curves of the proposed VAD method for different AR orders. (b) Comparison of ROC curves of the proposed method with those of reference methods in presence of a 60 Hz Additive Sinusoidal Component and SNR=5dB.

Table 6.8: Probability of Detection Versus Probability of False Alarm for Different SNR Levels and Different Methods (60 Hz Additive Sinusoidal Component)

	$P_{fa} = 5\%$			$P_{fa} = 10\%$			$P_{fa} = 20\%$			$P_{fa} = 40\%$		
SNR	0db	5db	10db	0db	5db	10db	0db	5db	10db	0db	5db	10db
$P_d$for Proposed Method	**88.06**	**88.91**	**90.09**	**90.51**	**91.04**	**92.32**	**91.26**	**93.18**	**94.78**	**93.92**	**94.56**	**95.10**
$P_d$for Ramirez Method	44.99	45.84	49.89	49.79	51.39	54.48	57.57	58.32	61.30	69.08	69.72	72.28
$P_d$for Sohn Method	12.79	18.55	29.96	21.54	29.10	39.98	36.67	42.22	52.67	52.03	55.01	60.45
$P_d$for Chang Method	25.59	36.46	41.15	36.03	44.24	49.36	47.55	56.08	61.62	61.09	65.67	66.74
$P_d$for Shin Method	34.33	39.02	46.06	44.14	48.51	54.48	55.33	60.13	63.22	64.61	66.31	69.19

6.5 Conclusions

We have presented a new VAD based on AR(p)-GARCH $(1,1)$ modeling of the speech signal. We also introduced a novel procedure based on the ML estimation method for parameter estimation of the AR(p)$-$GARCH $(1,1)$ model in presence of additive noise. We presented an adaptive version of parameter estimation method, namely, the RML method. In this method, upon receiving a new sample, we updated the likelihood function together with its gradient vector and used the steepest descent method to numerically find the maximum of the log-likelihood function. The proposed VAD is obtained by comparing the likelihood ratio to a threshold. The likelihood ratio of the observations was computed using the estimated variance of the speech signal, which was obtained by the RML algorithm. We used the IMCRA method for estimating the variance of the additive noise, which can be used under nonstationary conditions. This enabled our VAD to follow variations in the variance of the additive noise. Simulation results have demonstrated the high performance of the proposed method and particularly its advantage in nonstationary noise environments.

Appendix: Computation of the Gradient Vector

In this apendix, we compute the gradient of the pdf of the current observation conditioned on past observations, i.e., $\mathbf{g}_t = \nabla f\left(y_t|y_1^{t-1};\boldsymbol{\theta}\right)$. To find the pdf of the current observation conditioned on past observations, i.e., $f\left(y_t|y_1^{t-1};\boldsymbol{\theta}\right)$, note that the sequence of the clean signal is independent of the noise sequence; hence, using (6.11) and (6.12) we have

$$\left(y_t = x_t + n_t|y_1^{t-1};\boldsymbol{\theta}\right) \sim \mathcal{N}\left(\sum_{i=1}^{p}\alpha_i\hat{x}_{t-i}, \hat{\sigma}^2_{t|t-1}+\sigma^2\right) \tag{6.36}$$

or equivalently

$$\log f\left(\left(y_t|y_1^{t-1};\boldsymbol{\theta}\right)\right) = \log\frac{1}{\sqrt{2\pi}} - \frac{1}{2}\log\left(\hat{\sigma}^2_{t|t-1}+\sigma^2\right) - \frac{1}{2}\frac{\left(y_t-\sum_{i=1}^{p}\alpha_i\hat{x}_{t-i}\right)^2}{\hat{\sigma}^2_{t|t-1}+\sigma^2}. \tag{6.37}$$

For ease of notation, we define three intermediate variables as follows

$$\begin{aligned}\xi_{\alpha_i} &= \frac{\partial u_{t-1}}{\partial \alpha_i} \\ &= -2\left(y_{t-1} - \sum_{i=1}^{p} \alpha_i \hat{x}_{t-1-i}\right)\left(\frac{\hat{\sigma}^2_{t-1|t-2}}{\hat{\sigma}^2_{t-1|t-2} + \sigma^2_t}\right)^2 \hat{x}_{t-i-1} \qquad (6.38)\\ \chi &= \frac{\partial \log\left(f\left(y_t|y_0^{t-1};\boldsymbol{\theta}\right)\right)}{\partial \hat{\sigma}^2_{t|t-1}} = \frac{\partial \log\left(f\left(y_t|y_0^{t-1};\boldsymbol{\theta}\right)\right)}{\partial \sigma^2} \\ &= \left(\frac{\hat{\sigma}^2_{t|t-1} + \sigma^2_t - \left(y_t - \sum_{i=1}^{p} \alpha_i \hat{x}_{t-i}\right)^2}{\left(\hat{\sigma}^2_{t|t-1} + \sigma^2_t\right)^2}\right), \qquad (6.39)\end{aligned}$$

and

$$\begin{aligned}\chi_{\alpha_i} &= \frac{\partial \chi}{\partial \alpha_i} \\ &= -\frac{y_t - \sum_{i=1}^{p} \alpha_i \hat{x}_{t-i}}{\left(\hat{\sigma}^2_{t|t-1} + \sigma^2_t\right)^2} x_{t-i} - \frac{1}{2}\beta_1 \xi_{\alpha_i} \frac{2\left(y_t - \sum_{i=1}^{p} \alpha_i \hat{x}_{t-i}\right) - \left(\sigma^2_{t|t-1} + \sigma^2_t\right)}{\left(\hat{\sigma}^2_{t|t-1} + \sigma^2_t\right)^3}. \qquad (6.40)\end{aligned}$$

The gradient of the log-likelihood function of the current observation conditioned on past observations, i.e., $\mathbf{g}_t = \nabla \log\left(f\left(y_t|y_0^{t-1};\boldsymbol{\theta}\right)\right)$ can be computed as follows

$$\begin{aligned}[g_{t,1}, g_{t,2}, g_{t,3}]^T &= \chi \mathbf{a} \\ g_{t,i+3} &= \frac{\partial}{\partial \alpha_i}\left(\log\left(f\left(y_t|y_0^{t-1};\boldsymbol{\theta}\right)\right)\right) \\ &= \frac{y_t - \sum_{i=1}^{p} \alpha_i \hat{x}_{t-i}}{\hat{\sigma}^2_{t|t-1} + \sigma^2_t} \hat{x}_{t-i} + \chi \beta_1 \xi_{\alpha_i}; i = 1, 2, ..., p \qquad (6.41)\end{aligned}$$

where $g_{t,i}$ is the i-th element of vector $\mathbf{g}_t$ and $\mathbf{a} = \left[1, u_{t-1}, \hat{\sigma}^2_{t-1|t-2}\right]^T$.

Chapter 7

Two Dimensional Noncausal AR-ARCH Model: Stationary Conditions, Parameter Estimation and its Application to Anomaly Detection

7.1 Introduction

Image anomaly detection refers to the problem of finding regions in an image which do not conform to the expected behavior. The characteristic of the image and the type of anomalies are application dependent. Anomaly detection has a wide range of applications, such as silicon wafer defect detection [139, 140], medical imaging [113] and sea-mine detection in side-scan sonar images [91] just to name a few. In sea-mine detection, lethal targets must be detected with nearly one hundred percent reliability. False detections are not disastrous but might slow down the demining process. Every anomaly detection algorithm consists of some or all of the following stages: selection of an appropriate feature space; choosing an appropriate statistical model which represents the image background and selection of a detection algorithm.

A proper selection of a feature space, which allows distinction of anomalies from the background, is an important part of an anomaly detection algorithm. Features can be extracted from the image pixels themselves or from the image after passing through a transform. Kazantsev et al. [67], introduced a feature space based on two circular concentric windows W_1 and W_2 with

radii R_1 and R_2 $(R_1 < R_2)$, respectively. A similar approach was taken by Schweizer and Moura [104], where two concentric rectangles serve as the moving window. In these methods the features are extracted directly from the image itself. Features can be also extracted from the image in a transformed domain. Lain et al. [73] used a dyadic wavelet transform in mammography to emphasize mammographic features while reducing the noise. Strickland and Hahn [116] used an undecimated wavelet transform for detection of Gaussian objects in Markov noise. Xia et al. [131] used the wavelet transform to insert an undetectable watermark into digital imagery. Noibar et al. [91] used undecimated discrete wavelet transform (DWT) for anomaly detection in sonar images.

Once we have extracted our feature either from the image itself or from transformed image, we must find a statistical model to describe the background in the selected feature space domain. A survey of current literature shows that the most popular models for the background are Gaussian and its extensions. These models are used mostly because of their mathematical tractability. Ashton [11] performed a sub-pixel anomaly detection in multispectral infrared imagery using a Gaussian distribution. Stein et al. [114] used a Gaussian mixture model (GMM) for modeling hyper-spectral imagery. Other extensions such as linear mixing model (LMM) and Gauss-Markov random field (GMRF) are used by several authors for modeling the background. A review of multi-resolution Markov models for signal and image processing can be found in [130]. The generalized autoregressive conditional heteroscedasticity (GARCH) model was first introduced by Bollerslev [19] as an extension of the autoregressive conditional heteroscedasticity (ARCH) model developed by Engle [42] to model econometric data. Since then, many researchers have extended and used these models in several speech and image processing applications. Cohen [25], modeled the speech signal in the short time Fourier transform (STFT) domain as a complex GARCH process and used this model for speech enhancement. AR-GARCH model was utilized for modeling speech signal in the time domain and for developing voice activity detection (VAD) algorithms [84, 112]. Abdolahi et al. [2], used the parameters of the GARCH model for speech recognition in Persian isolated digits. Amirmazlaghani et al. [10], used two dimensional GARCH model for speckle suppression in SAR images. Two dimensional GARCH model is also used in image denoising [9]. Noiboar and Cohen [91], used *causal* GARCH model for anomaly detection in sonar images. The causality assumption incorporated into the GARCH model in [91], is unnatural for images. Developing a

non-causal statistical model may lead to an improvement in detection performance by reducing the dependency of the detection procedure on image orientation.

In [82], we presented an anomaly detection method in sonar images based on noncausal AR-ARCH model. The background of the sonar image in the wavelet domain was modeled by a noncausal AR-ARCH model. Matched subspace detector (MSD) was used for detecting the anomaly in the image. In MSD it is assumed that the anomalies are within a subspace. This subspace is assumed to be known or can be estimated using training data.

In this chapter, we provide sufficient stationary conditions for the model and propose an effective least squares method for estimating the model parameters. This estimator, which is shown to be asymptotically consistent, is obtained by solving two sets of linear equations and have a closed-form expression. We also present the detection algorithm in more details. This algorithm is based on the noncausal AR-ARCH modeling of the background and MSD.

This chapter is organized as follows. In Section 7.2, we introduce a two dimensional noncausal ARCH model, provide sufficient stationary conditions and develop a novel technique for estimating the parameters of this model. We also show in Appendix that this estimator is asymptotically consistent. In Section 7.3, we introduce our anomaly detection algorithm, which is based on noncausal autoregressive ARCH model and MSD. This section is a detailed description of the method previously proposed by the authors in [82]. In Section 7.4, the performances of the parameter estimation and anomaly detection are evaluated using simulations. We conclude the chapter in Section 7.5.

7.2 Noncausal ARCH Model

7.2.1 Two dimensional Noncausal ARCH Model and its Stationary Conditions

We define a two dimensional noncausal ARCH(p,q) model as follows

$$x(t_1,t_2) = \sigma(t_1,t_2)\varepsilon(t_1,t_2), \tag{7.1}$$

$$\sigma^2(t_1,t_2) = c_0 + \sum_{i=-p}^{p}\sum_{j=0}^{q} a_{i,j}\left(x^2(t_1-i,t_2-j)+x^2(t_1+i,t_2+j)\right), \tag{7.2}$$

$$c_0 > 0 \qquad a_{i,j} \geq 0, \tag{7.3}$$

where c_0 and a_{ij}'s are the two dimensional noncausal ARCH parameters, $\forall i \geq 0 \; a_{i,0} = 0$, p and q are the model orders in the horizontal and vertical directions, respectively. $\varepsilon(t_1,t_2)$'s are zero mean independent identically distributed (IID) random variables with unit variance and $\sigma^2(t_1,t_2)$'s are called the conditional variances. The constraints (7.3) guarantee the positiveness of σ^2. These equations simply state that each pixel in the image (i.e. $x(t_1,t_2)$) is a random variable whose conditional variance (conditioned on the neighboring pixels) is a weighted sum of the squared value of the neighboring pixels. The neighborhood is determined by p and q. This definition is explained graphically in Figure 7.1.

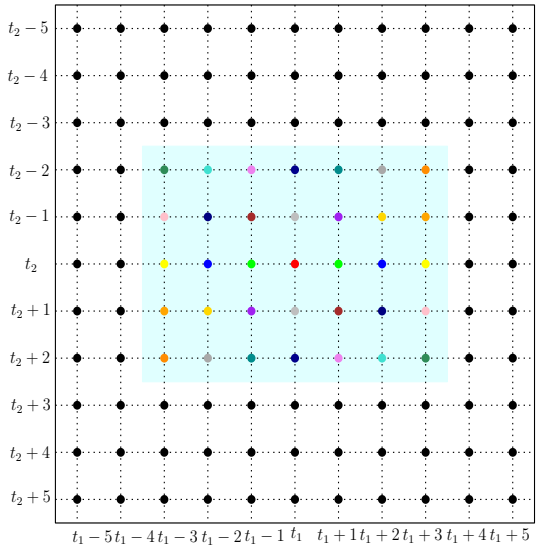


Figure 7.1: The conditional variance of the centered pixel is a weighted mean of the squared value of the neighboring pixels where the weights are symmetric (i.e. pixels dotted with the same color have the same weight).

This figure shows that the conditional variance of the centered pixel is a weighted mean of the squared value of the neighboring pixels where the weights are symmetric (i.e. pixels dotted with the same color have the same weight). This condition is applied to the definition of the model in order to make it identifiable. More specifically, since our parameter estimation method is based on the LS method proposed in [135] for estimating the parameters of the noncausal AR model, and a noncausal AR model is identifiable by the LS method if and only if the weights are symmetric, we ought to assume that the weights are symmetric. Two dimensional noncausal AR model represents the gray scale level at a specific pixel, as a linear combination of the gray scale levels of neighboring pixels and an additive white noise. This model has many applications in

image processing and analysis. For example, it can be used for the design of image restoration algorithms [62, 122], for multidimensional spectral estimation [80] , and for texture analysis and synthesis [124].

Theorem:

A two dimensional noncausal ARCH model is ergodic with finite second order moment if

$$\sup_{t_1,t_2} \varepsilon^2(t_1,t_2) = \kappa < \infty \tag{7.4}$$

$$\sum_{i=-p}^{p} \sum_{j=0}^{q} a_{i,j} < \frac{1}{2\kappa} \tag{7.5}$$

Proof:

We define a semi metric space (X,d) as

$$X = \left\{ x(t_1,t_2) | \lim_{N\to\infty} \frac{1}{(2N+1)^2} \sum_{i=-N}^{N} \sum_{j=-N}^{N} x^2(i,j) < \infty \right\} \tag{7.6}$$

with the following semi metric

$$d(x,y) = \lim_{N\to\infty} \frac{1}{(2N+1)^2} \sum_{i=-N}^{N} \sum_{j=-N}^{N} |x^2(i,j) - y^2(i,j)|. \tag{7.7}$$

Let $T(x(t_1,t_2))$ be the following mapping

$$T\left(x(t_1,t_2)\right) = \left(c_0 + \sum_{i=-p}^{p} \sum_{j=0}^{q} a_{i,j} \left(x^2(t_1-i,t_2-j) + x^2(t_1+i,t_2+j) \right) \right)^{\frac{1}{2}} \varepsilon(t_1,t_2). \tag{7.8}$$

If we show that (X,d) is a complete space and $T(x(t_1,t_2))$ is a contraction mapping from X to itself, then using Banach's fixed point theorem $x(t_1,t_2) = T(x(t_1,t_2))$ has at least one fixed point. This means that $x(t_1,t_2)$ is ergodic with finite second order moment and hence stationary.

To show that (X,d) is a complete space we must show that every Cauchy sequence in X converges to an element in X with respect to the metric d. This can be done in a similar way of proving completeness of ℓ^2 which can be found in every classical textbook on functional analysis. See for example [69]. In order to show that $T(x(t_1,t_2))$ is a contraction mapping from X to itself, first we must show that if $x(t_1,t_2) \in X$ then $y(t_1,t_2) = T(x(t_1,t_2)) \in X$. Note that

$\forall N \in \mathbb{N}$ we have

$$\begin{aligned}
\alpha_N &= \frac{1}{(2N+1)^2} \sum_{t_1=-N}^{N} \sum_{t_2=-N}^{N} y^2(i,j) \\
&= \frac{1}{(2N+1)^2} \sum_{i=-N}^{N} \sum_{j=-N}^{N} c_0 \varepsilon^2(t_1,t_2) \\
&+ \frac{1}{(2N+1)^2} \sum_{i=-p}^{p} \sum_{j=0}^{q} a_{i,j} \sum_{t_1=-N}^{N} \sum_{t_2=-N}^{N} \varepsilon^2(t_1,t_2) \left(x^2(t_1-i,t_2-j) + x^2(t_1+i,t_2+j) \right) \\
&\leq \frac{\kappa}{(2N+1)^2} \sum_{i=-N}^{N} \sum_{j=-N}^{N} c_0 + \frac{\kappa}{(2N+1)^2} \sum_{i=-p}^{p} \sum_{j=0}^{q} a_{i,j} \sum_{t_1=-N}^{N} \sum_{t_2=-N}^{N} x^2(t_1-i,t_2-j) + x^2(t_1+i,t_2+j) \\
&= \kappa c_0 + \frac{\kappa}{(2N+1)^2} \sum_{i=-p}^{p} \sum_{j=0}^{q} a_{i,j} \sum_{t_1=-N}^{N} \sum_{t_2=-N}^{N} x^2(t_1-i,t_2-j) + x^2(t_1+i,t_2+j). \qquad (7.9)
\end{aligned}$$

Taking a limit from both sides of the last inequality (7.9) and considering the fact that $x \in X$, it can easily be verified that

$$\lim_{N\to\infty} \alpha_N < \infty,$$

hence, T is a mapping from X to X.

For a mapping to be a contraction mapping on X, there must exist a constant $\beta < 1$ such that $d(Tx,Ty) < \beta d(x,y)$ for all $x,y \in X$. Now note that

$$\begin{aligned}
d_N(Tx,Ty) &= \frac{1}{(2N+1)^2} \sum_{t_1=-N}^{N} \sum_{t_2=-N}^{N} |x^2(i,j) - y^2(i,j)| \\
&= \frac{1}{(2N+1)^2} \sum_{i=-p}^{p} \sum_{j=0}^{q} a_{i,j} \sum_{t_1=-N}^{N} \sum_{t_2=-N}^{N} \varepsilon^2(t_1,t_2) |x^2(t_1-i,t_2-j) - y^2(t_1-i,t_2-j)| \\
&+ \frac{1}{(2N+1)^2} \sum_{i=-p}^{p} \sum_{j=0}^{q} a_{i,j} \sum_{t_1=-N}^{N} \sum_{t_2=-N}^{N} \varepsilon^2(t_1,t_2) |x^2(t_1+i,t_2+j) - y^2(t_1+i,t_2+j)|.
\end{aligned} \qquad (7.10)$$

Using a similar approach we used to obtain (7.9), it can be shown that

$$d(Tx,Ty) = \lim_{N\to\infty} d_N(Tx,Ty) \leqslant \left(2\kappa \sum_{i=-p}^{p} \sum_{j=0}^{q} a_{i,j} \right) d(x,y). \qquad (7.11)$$

Hence, $\beta = \left(2\kappa \sum_{i=-p}^{p} \sum_{j=0}^{q} a_{i,j} \right)$ and a sufficient stationary condition is

$$\beta < 1 \Rightarrow \sum_{i=-p}^{p} \sum_{j=0}^{q} a_{i,j} < \frac{1}{2\kappa}. \qquad (7.12)$$

Note that the second condition can be interpreted as a condition on the probability distribution function (pdf) of $\varepsilon(t_1,t_2)$ to be compactly supported.

7.2.2 Parameter Estimation of Two Dimensional Noncausal ARCH Model

Parameter estimation is an important part of any model-based signal processing algorithm. There are several parameter estimation methods for ARCH and GARCH models, for example quasi maximum likelihood (QML) [129], two stage least squares (TSLS) [20], constrained two stage least squares (CTSLS) [86] and multivariate two stage least squares (MTSLS) [87]. Since ARCH and GARCH models are mostly used in econometrics, where processes can be assumed causal, to the best of our knowledge, there was no attempt to define noncausal (G)ARCH models and finding a parameter estimation method for the noncausal case. Upon defining a two dimensional noncausal ARCH model in the previous section, in this section, we develop a novel parameter estimation method for estimating the parameters of this model. The proposed method is mainly based on MTSLS and uses the same approach utilized in [135] for parameter estimation of the two dimensional noncausal AR model.

Let $y(t_1,t_2) = x^2(t_1,t_2)$. Then from the definition of the two dimensional noncausal ARCH model (7.1)-(7.2), we have

$$\begin{aligned} y(t_1,t_2) &= c_0 + \sum_{i=-p}^{p}\sum_{j=0}^{q} a_{i,j}(y(t_1-i,t_2-j)+y(t_1+i,t_2+j)) + \sigma^2(t_1,t_2)(\varepsilon^2(t_1,t_2)-1) \\ &= c_0 + \sum_{i=-p}^{p}\sum_{j=0}^{q} a_{i,j}(y(t_1-i,t_2-j)+y(t_1+i,t_2+j)) + \sigma^2(t_1,t_2)\eta(t_1,t_2). \end{aligned} \tag{7.13}$$

Ignoring the dependence of $\sigma^2(t_1,t_2)\eta(t_1,t_2)$ on the parameters, (7.13) is similar to a two dimensional noncausal AR process. We utilize a similar approach employed in [135] for parameter estimation. Note that $y(t_1,t_2)$ can be divided into the following two respectively causal and anti-causal processes

$$\frac{1}{2}y(t_1,t_2) = \frac{c_0}{2} + \sum_{i=-p}^{p}\sum_{j=0}^{q} a_{i,j}y(t_1-i,t_2-j) + \sigma^2(t_1,t_2)\eta_L(t_1,t_2) \tag{7.14}$$

$$\frac{1}{2}y(t_1,t_2) = \frac{c_0}{2} + \sum_{i=-p}^{p}\sum_{j=0}^{q} a_{i,j}y(t_1+i,t_2+j) + \sigma^2(t_1,t_2)\eta_U(t_1,t_2), \tag{7.15}$$

where $\eta_L(t_1,t_2)$ and $\eta_U(t_1,t_2)$ are the corresponding causal and anti-causal prediction errors such that $\eta_L(t_1,t_2)+\eta_U(t_1,t_2) = \eta(t_1,t_2)$. As in [135], we can assume that $\eta_L(t_1,t_2)$ and $\eta_U(t_1,t_2)$ are uncorrelated. Furthermore, It can also be assumed that $\eta_L(t_1,t_2)$ and $\eta_U(t_1,t_2)$ are uncorrelated with $y(t_1 \pm i, t_2 \pm j) : -p \le i \le p, 0 \le j \le q$. In order to check that if these

assumptions hold, we simulate a noncausal ARCH$(1,1)$ model and computed the correlation between η, η_L, η_U, and y. In this experiment, we simulated a noncausal ARCH$(1,1)$ process of the size 200×200 using Banach fixed point theorem with 15 iterations. The model parameters where set to

$$c_0 = 1, \quad a_{-1,-1} = 0.03, \quad a_{0,-1} = 0.04, \quad a_{1,-1} = 0.05, \quad a_{-1,0} = 0.06.$$

The process noise $\varepsilon(t_1,t_2)$ was chosen to be a sequence of IID random variables driven from uniform distribution having zero mean and unity variance (i.e. $\varepsilon(t_1,t_2) \sim U\left(\left[-\sqrt{3},\sqrt{3}\right]\right)$). The sample cross-correlation of column stack of η_L and y, η_U and y, η and y, and η_L and η_U, are depicted in Figures 7.2 (a) through (d), respectively.

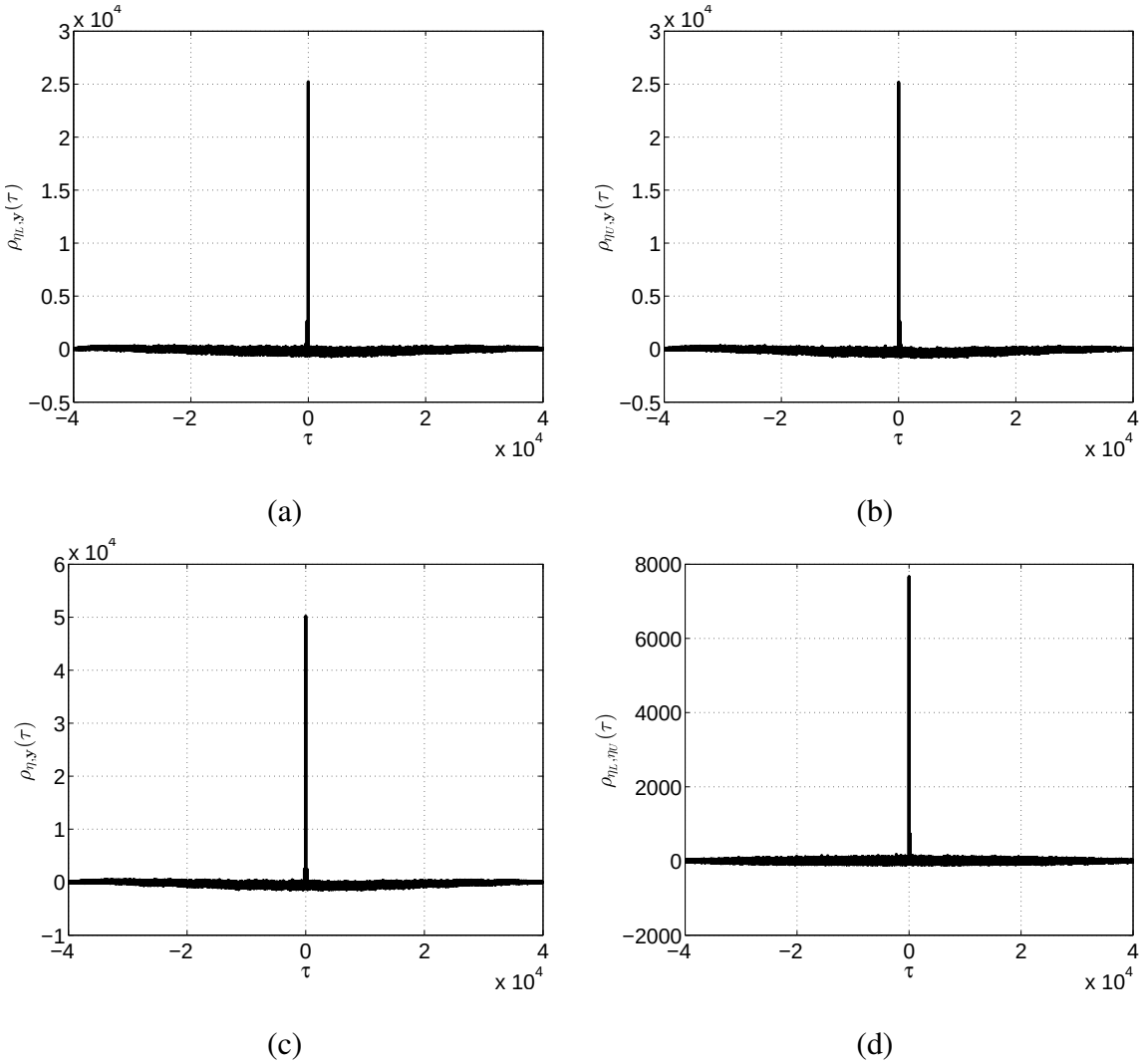


Figure 7.2: The sample cross-correlation of: (a) η_L and y, (b) η_U and y, (c) η and y, and (d) η_L and η_U.

From these figures, it is obvious that these process are uncorrelated for any lag greater than zero. Hence, the parameters of the model can be estimated by minimizing the following cost function

$$C(\boldsymbol{a}) = \sum_{t_1=p+1}^{N-p} \sum_{t_2=q+1}^{M-q} \left(\frac{1}{2}y(t_1,t_2) - \boldsymbol{a}^T \boldsymbol{y}_L(t_1,t_2)\right)^2 + \left(\frac{1}{2}y(t_1,t_2) - \boldsymbol{a}^T \boldsymbol{y}_U(t_1,t_2)\right)^2. \quad (7.16)$$

with respect to $\boldsymbol{a}$, where N and M are the number of available data in horizontal and vertical directions, respectively, and $(\cdot)^T$ is the transpose operator. The vector of parameters $\boldsymbol{a}$, $\boldsymbol{y}_L(t_1,t_2)$ and $\boldsymbol{y}_U(t_1,t_2)$ are defined as follows

$$\boldsymbol{a} = \left[c_0, a_{-p,0}...a_{-1,0}, a_{-p,1}...a_{p,1}, a_{-p,2}...a_{p,2}, ...a_{-p,q}, ...a_{p,q}\right]^T \quad (7.17)$$

$$\begin{aligned}\boldsymbol{y}_L(t_1,t_2) = [\tfrac{1}{2}, \quad & y(t_1-1,t_2)...y(t_1-p,t_2), y(t_1+p,t_2-1)...y(t_1-p,t_2-1), \\ & y(t_1+p,t_2-2)...y(t_1-p,t_2-2), ..., y(t_1+p,t_2-q)...y(t_1-p,t_2-q)]^T\end{aligned} \quad (7.18)$$

$$\begin{aligned}\boldsymbol{y}_U(t_1,t_2) = [\tfrac{1}{2}, \quad & y(t_1+1,t_2)...y(t_1+p,t_2), y(t_1-p,t_2+1)...y(t_1+p,t_2+1), \\ & y(t_1-p,t_2+2)...y(t_1+p,t_2+2), ..., y(t_1-p,t_2+q)...y(t_1+p,t_2+q)]^T.\end{aligned} \quad (7.19)$$

Sice it is difficult to obtain statistical properties of this estimator, we propose the following suboptimal parameter estimation method

$$\hat{\boldsymbol{a}}_{pr} = \frac{1}{2}\left(\arg\min_{\boldsymbol{a}} \sum_{t_1=p+1}^{N-p} \sum_{t_2=q+1}^{M-q} \left(\frac{1}{2}y(t_1,t_2) - \boldsymbol{a}^T \boldsymbol{y}_L(t_1,t_2)\right)^2 + \arg\min_{\boldsymbol{a}} \left(\frac{1}{2}y(t_1,t_2) - \boldsymbol{a}^T \boldsymbol{y}_U(t_1,t_2)\right)^2\right). \quad (7.20)$$

which, as we show latter, is asymptotically unbiased and consistent. The minimum of each these cost functions in (7.20) can be found by setting the gradient of the corresponding cost function with respect to the vector of parameters to zero. These cost functions have quadratic forms in parameters and therefore are convex and have a unique minimum. Hence a *primary* estimate of the parameters is obtained as follows

$$\hat{\boldsymbol{a}}_{pr}^T = \frac{1}{2}\left(\boldsymbol{x}_L^T \boldsymbol{R}_L^{-1} + \boldsymbol{x}_U^T \boldsymbol{R}_U^{-1}\right), \quad (7.21)$$

where

$$\boldsymbol{R}_L = 2\sum_{t_1=p+1}^{N-p}\sum_{t_2=q+1}^{M-q} \boldsymbol{y}_L(t_1,t_2)\boldsymbol{y}_L^T(t_1,t_2) \tag{7.22}$$

$$\boldsymbol{R}_U = 2\sum_{t_1=p+1}^{N-p}\sum_{t_2=q+1}^{M-q} \boldsymbol{y}_U(t_1,t_2)\boldsymbol{y}_U^T(t_1,t_2) \tag{7.23}$$

$$\boldsymbol{x}_L = \sum_{t_1=p+1}^{N-p}\sum_{t_2=q+1}^{M-q} y(t_1,t_2)\boldsymbol{y}_L(t_1,t_2) \tag{7.24}$$

$$\boldsymbol{x}_U = \sum_{t_1=p+1}^{N-p}\sum_{t_2=q+1}^{M-q} y(t_1,t_2)\boldsymbol{y}_U(t_1,t_2). \tag{7.25}$$

We denote this estimator by $\hat{\boldsymbol{a}}_{pr}$ where the subscript pr stands for 'preliminary'. This is due to the fact that in deriving this estimator we have not considered the dependency of $\sigma^2(t_1,t_2)\eta(t_1,t_2)$ on the parameters. Using (7.2) and this preliminary estimate of the parameters, we can estimate the conditional variance $\sigma^2(t_1,t_2)$, in the following way

$$\hat{\sigma}^2(t_1,t_2) = \hat{c}_0^{pr} + \sum_{i=-p}^{p}\sum_{j=0}^{q} \hat{a}_{i,j}^{pr}(y(t_1-i,t_2-j)+y(t_1+i,t_2+j)) = \hat{\boldsymbol{a}}_{pr}^T\left(\boldsymbol{y}_L(t_1,t_2)+\boldsymbol{y}_U(t_1,t_2)\right), \tag{7.26}$$

where $\hat{c}_0^{pr}$ and $\hat{a}_{i,j}^{pr}$ are the preliminary estimates of the parameters obtained in (7.21). It follows from (7.13) that

$$\frac{y(t_1,t_2)}{\hat{\sigma}^2(t_1,t_2)} \approx \frac{c_0}{\hat{\sigma}^2(t_1,t_2)} + \sum_{i=-p}^{p}\sum_{j=0}^{q} a_{i,j}\left(\frac{y(t_1-i,t_2-j)}{\hat{\sigma}^2(t_1,t_2)} + \frac{y(t_1+i,t_2+j)}{\hat{\sigma}^2(t_1,t_2)}\right) + (\varepsilon^2(t_1,t_2)-1). \tag{7.27}$$

Note that the two sides of (7.27) are not exactly equal because we have canceled out $\sigma^2(t_1,t_2)$ by its estimate, i.e. $\hat{\sigma}^2(t_1,t_2)$. Now, we use the above equation to find our final estimator. If we ignore the approximation used in deriving (7.27), we can derive the LS estimate of the parameters (i.e. $\boldsymbol{a}$) by solving the following optimization problem

$$\hat{\boldsymbol{a}} = \frac{1}{2}\left(\arg\min_{\boldsymbol{a}} \sum_{t_1=p+1}^{N-p}\sum_{t_2=q+1}^{M-q}\left(\frac{1}{2}\overline{y}(t_1,t_2) - \boldsymbol{a}^T\overline{\boldsymbol{y}}_L(t_1,t_2)\right)^2 + \arg\min_{\boldsymbol{a}}\left(\frac{1}{2}\overline{y}(t_1,t_2) - \boldsymbol{a}^T\overline{\boldsymbol{y}}_U(t_1,t_2)\right)^2\right), \tag{7.28}$$

where

$$\overline{y}(t_1,t_2) = \frac{y(t_1,t_2)}{\hat{\sigma}^2(t_1,t_2)} \tag{7.29}$$

$$\overline{\boldsymbol{y}}_L(t_1,t_2) = \frac{\boldsymbol{y}_L(t_1,t_2)}{\hat{\sigma}^2(t_1,t_2)} \tag{7.30}$$

$$\overline{\boldsymbol{y}}_U(t_1,t_2) = \frac{\boldsymbol{y}_U(t_1,t_2)}{\hat{\sigma}^2(t_1,t_2)}. \tag{7.31}$$

Again, since each of the cost function is a quadratic function of the vector $\boldsymbol{a}$, they are convex and have a unique minimum which can be found by setting the derivative of $\overline{C}(\boldsymbol{a})$ to zero. Hence, our final estimator is obtained as follows

$$\hat{\boldsymbol{a}}^T = \frac{1}{2}\left(\overline{\boldsymbol{x}}_L^T \overline{\boldsymbol{R}_L}^{-1} + \overline{\boldsymbol{x}}_U^T \overline{\boldsymbol{R}_L}^{-1}\right), \tag{7.32}$$

where

$$\overline{\boldsymbol{R}_L} = 2\sum_{t_1=p+1}^{N-p}\sum_{t_2=q+1}^{M-q} \overline{\boldsymbol{y}}_L(t_1,t_2)\overline{\boldsymbol{y}}_L^T(t_1,t_2) \tag{7.33}$$

$$\overline{\boldsymbol{R}_U} = 2\sum_{t_1=p+1}^{N-p}\sum_{t_2=q+1}^{M-q} \overline{\boldsymbol{y}}_U(t_1,t_2)\overline{\boldsymbol{y}}_U^T(t_1,t_2) \tag{7.34}$$

$$\overline{\boldsymbol{x}}_L = \sum_{t_1=p+1}^{N-p}\sum_{t_2=q+1}^{M-q} \overline{y}(t_1,t_2)\overline{\boldsymbol{y}}_L(t_1,t_2) \tag{7.35}$$

$$\overline{\boldsymbol{x}}_U = \sum_{t_1=p+1}^{N-p}\sum_{t_2=q+1}^{M-q} \overline{y}(t_1,t_2)\overline{\boldsymbol{y}}_U(t_1,t_2). \tag{7.36}$$

We denote this estimator as two stage least squares estimator (TSLS). In appendix, we show that under some moment conditions besides ergodicity, this estimator is asymptotically unbiased and consistent. It should be noted that the TSLS estimator does not guarantee that the estimated model is stationary. However, this method can be easily modified (by solving the least squares problems under stationary conditions) such that the estimated model be stationary, as was done for the univariate causal ARCH case in [86].

7.3 Anomaly Detection Based on Noncausal AR-ARCH Model

In this section, we introduce our anomaly detection algorithm which is based on noncausal AR-ARCH modeling of the image background in the wavelet domain. This section is mainly based

on our previous work on anomaly detection [82]. We use a matched subspace detector (MSD) proposed in [101] along with noncausal AR introduced in [135] and noncausal ARCH model introduced in the previous section. We assume that the anomalies in the images are so rare that their influence on the parameter estimation algorithm can be ignored. The proposed algorithm is an extension of the method proposed in [91]. In [91], the authors introduced an anomaly detection algorithm based on causal AR-GARCH modeling of the images in the wavelet domain. Causality of the model in [91] leads to dependence of the anomaly detection procedure on the orientation of the image. This means that the algorithm is sensitive to the orientation of the image, so in order to use this method one must consider all four orientations of the image which is a time consuming process. Another disadvantage of the procedure proposed in [91] is its high computational load. That method uses causal AR-GARCH model in the wavelet domain, and the parameters of the GARCH part are estimated using a maximum likelihood (ML) method, which has a very high computational load, since the maximum of the likelihood function is found by a search method. Our method overcomes both of these disadvantages. Using the proposed noncausal model, the sensitivity of our method to image orientation is reduced significantly and the computational load is decreased considerably using the proposed TSLS method for parameter estimation of the noncausal ARCH model.

In what follows, we introduce our noncausal AR-ARCH model for the image background in the wavelet domain. This model is to some extent similar to the model used in [91] with a major difference. The model used in [91] is a causal model whereas our model is noncausal. Let $z(t_1,t_2)$ be the original image. Using the two dimensional undecimated wavelet transform [128], we obtain a set of $2L+1$ images where L is the depth of the wavelet transform. These $2L+1$ images are obtained as follows. The undecimated wavelet transform yields four sub-band images at every analysis level. These four sub-band images are labeled by d^i_{LH}, d^i_{HL}, d^i_{HH} and s^i_{LL}, where the superscript i represents the layer index and the subscripts L and H stand for low-pass and high-pass filtering, respectively. d labels a detail sub-band, s represents the smooth sub-band, and the superscript ℓ specifies the analysis level. The undecimated wavelet transform yields a redundant representation. However, the same analysis and synthesis filters are used as in the undecimated wavelet transform, and since the transform preserves the spatial dimensions, it is easy to work with. Furthermore, the undecimated wavelet transform has an additional property, namely, translation invariance, which is important in the context of anomaly

detection. As in [116], we use the sub-band images of an undecimated wavelet transform to create a $2L+1$ feature images at every spatial location (t_1,t_2) as follows

$$y_{2i-1}(t_1,t_2) = d^i_{\mathrm{LH}}(t_1,t_2) + d^i_{\mathrm{HL}}(t_1,t_2) \quad ; \quad 1 \le i \le L \tag{7.37}$$

$$y_{2i}(t_1,t_2) = d^i_{\mathrm{HH}}(t_1,t_2) \quad ; \quad 1 \le i \le L \tag{7.38}$$

$$y_{2L+1}(t_1,t_2) = s^{2L+1}_{\mathrm{LL}}(t_1,t_2). \tag{7.39}$$

Let $y_\ell(t_1,t_2)$; $1 \le \ell \le 2L+1$ be the ℓ-th feature image in the wavelet domain, obtained by (7.37) through (7.39). We assume that the background image is a noncausal AR-ARCH process defined as follows

$$y_\ell(t_1,t_2) = \sum_{i=-r}^{r}\sum_{j=0}^{s} b^\ell_{i,j}\left(y_\ell(t_1-i,t_2-j) + y_\ell(t_1+i,t_2+j)\right) + x_\ell(t_1,t_2) \tag{7.40}$$

$$x_\ell(t_1,t_2) = \sigma_\ell(t_1,t_2)\varepsilon_\ell(t_1,t_2) \tag{7.41}$$

$$\sigma^2_\ell(t_1,t_2) = c^\ell_0 + \sum_{i=-p}^{p}\sum_{j=0}^{q} a^\ell_{i,j}\left(x^2_\ell(t_1-i,t_2-j) + x^2_\ell(t_1+i,t_2+j)\right), \tag{7.42}$$

where b^ℓ_{ij}'s are parameters of the AR part, $\forall i \ge 0$ $b^\ell_{i,0} = 0$ and r and s are the order of AR model in the horizontal and vertical directions, respectively. c_0 and a_{ij}'s are the parameters of the ARCH part and $\forall i \ge 0$ $a_{i,0} = 0$, p and q are the order of ARCH model in horizontal and vertical directions, respectively. $\varepsilon_\ell(t_1,t_2)$'s are zero mean independent identically distributed (IID) random variables. We have defined our two dimensional noncausal AR-ARCH model based on the definition of the two dimensional noncausal AR model [135] extensively used in image signal processing and the two dimensional noncausal ARCH model introduced in Section 7.2.

Now we propose a method for parameter estimation of the two dimensional noncausal AR-ARCH model. Considering the fact that the ARCH process is white, the parameters of the AR part can be easily estimated with the method proposed in [135]. Let this estimate of the parameters of the AR part be denoted by $\hat{b}^\ell_{ij}$. Substituting this estimate of the parameters in (7.40), the residuals of the AR model (i.e. $x_\ell(t_1,t_2)$) can be estimated as follows

$$\hat{x}_\ell(t_1,t_2) = y_\ell(t_1,t_2) - \sum_{i=-p}^{r}\sum_{j=0}^{s} \hat{b}^\ell_{i,j}(y_\ell(t_1-i,t_2-j) + y_\ell(t_1+i,t_2+j)). \tag{7.43}$$

Using this estimate of the residuals, the parameters of the ARCH part can be estimated using the TSLS method introduced in the previous section. Substituting this estimate of the parameters

of the ARCH part and the estimate of the residuals (i.e. $\hat{x}^{\ell}(t_1,t_2)$) in (7.42), the estimate of the conditional variance is obtained as follows

$$\hat{\sigma}^2_{\ell}(t_1,t_2) = \hat{c}^{\ell}_0 + \sum_{i=-p}^{p}\sum_{j=0}^{q} \hat{a}^{\ell}_{i,j}(\hat{x}^2_{\ell}(t_1-i,t_2-j) + \hat{x}^2_{\ell}(t_1+i,t_2+j)), \tag{7.44}$$

where $\hat{c}^{\ell}_0$ and $\hat{a}^{\ell}_{i,j}$ are the estimates of the parameters of the ARCH part obtained by the TSLS method. The order of the model (i.e. p, q, r and s) can be selected by Akaike information criterion (AIC) or Bayesian information criterion (BIC). In this work we select the order of the model experimentally. A more detailed discussion on order selection needs further research and is beyond the scope of this chapter.

Using this estimate of the conditional variance, the anomaly detector is obtained as follows. For each pixel in each layer, we create a column vector $\hat{\boldsymbol{x}}_{\ell}(t_1,t_2)$ by column stacking an image chip of size $T_1 \times T_2$ of $\hat{x}_{\ell}(t_1,t_2)$ centered around the pixel (t_1,t_2) in the ℓ-th layer. Assume that there exists no interference and let $\psi_{\ell}(t_1,t_2)$ be vector locating the anomaly within its subspace $\boldsymbol{H}_{\ell} = \text{span}(\boldsymbol{H}_{\ell})$. These matrices, i.e. $\boldsymbol{H}_{\ell}$; $1 \leq \ell \leq 2L+1$, are obtained from a set of M training patches containing anomalies. Succinctly, suppose we are given a set of training images containing mines. Each of these images is transformed to the wavelet domain and for each image, $2L+1$ feature images are extracted using (7.37) through (7.39). In a specific layer ℓ, we used the corresponding M images to obtain the anomaly subspace in that layer. This task is done by first extracting all possible $T_1 \times T_2$ image chips (without overlap) from these M images, column stacking each of these patches and finding the K principle components using principle component analysis (PCA), see [91] for a more detailed discussion. Let $\boldsymbol{\sigma}^2_{\ell}(t_1,t_2)$ be a vector obtained by column stacking of an image chip of size $T_1 \times T_2$ of $\sigma^2_{\ell}(t_1,t_2)$ centered around the pixel (t_1,t_2) in the ℓ-th layer. Let $\Sigma_{\ell}(t_1,t_2)$ is a diagonal matrix whose diagonal is the vector $\boldsymbol{\sigma}^2_{\ell}(t_1,t_2)$. We define two hypotheses, $\mathcal{H}_0$ and $\mathcal{H}_1$, which respectively represent absence and presence of an anomaly as follows

$$\mathcal{H}_0 : \hat{\boldsymbol{x}}_{\ell}(t_1,t_2) = \Sigma_{\ell}(t_1,t_2)\varepsilon_{\ell}(t_1,t_2) \tag{7.45}$$

$$\mathcal{H}_1 : \hat{\boldsymbol{x}}_{\ell}(t_1,t_2) = \boldsymbol{H}_{\ell}\psi_{\ell}(t_1,t_2) + \Sigma_{\ell}(t_1,t_2)\varepsilon_{\ell}(t_1,t_2), \tag{7.46}$$

Under these two hypotheses, we assume that the sample conditional distribution of $\hat{\boldsymbol{x}}_{\ell}(t_1,t_2)$

is Gaussian with identical covariance matrices with different means, i.e.

$$\mathcal{H}_0 : \hat{\boldsymbol{x}}_\ell(t_1,t_2) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_\ell(t_1,t_2)) \tag{7.47}$$

$$\mathcal{H}_1 : \hat{\boldsymbol{x}}_\ell(t_1,t_2) \sim \mathcal{N}\left(\boldsymbol{H}_\ell \boldsymbol{\psi}^\ell(t_1,t_2), \boldsymbol{\Sigma}_\ell(t_1,t_2)\right) \tag{7.48}$$

Under this assumption, the log-likelihood ratio in the ℓ-th layer, is given by [91]

$$L_\ell(t_1,t_2) = \left((\boldsymbol{\Sigma}_\ell(t_1,t_2))^{-\frac{1}{2}} \hat{\boldsymbol{x}}_\ell(t_1,t_2)\right)^T P_{\boldsymbol{H}_\ell} \left((\boldsymbol{\Sigma}_\ell(t_1,t_2)^\ell)^{-\frac{1}{2}} \hat{\boldsymbol{x}}_\ell(t_1,t_2)\right) \tag{7.49}$$

where

$$P_{\boldsymbol{H}_\ell} = \boldsymbol{H}_\ell \left(\boldsymbol{H}_\ell^T \boldsymbol{H}_\ell\right)^{-1} \boldsymbol{H}_\ell^T \tag{7.50}$$

Since the true value of the conditional variance ($\boldsymbol{\Sigma}_\ell(t_1,t_2)$) is not available, by the generalized likelihood ratio test, it can be replaced by its estimate ($\widehat{\boldsymbol{\Sigma}}_\ell(t_1,t_2)$). Our final detector is given by comparing the generalized likelihood ratio with a predefined threshold, i.e.

$$L_{GLR}(t_1,t_2) = \sum_{\ell=1}^{2L+1} \left(\left(\widehat{\boldsymbol{\Sigma}}_\ell(t_1,t_2)\right)^{-\frac{1}{2}} \hat{\boldsymbol{x}}_\ell(t_1,t_2)\right)^T P_{\boldsymbol{H}_\ell} \left(\left(\widehat{\boldsymbol{\Sigma}}_\ell(t_1,t_2)\right)^{-\frac{1}{2}} \hat{\boldsymbol{x}}_\ell(t_1,t_2)\right) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} T_h \tag{7.51}$$

where $\widehat{\boldsymbol{\Sigma}}_\ell(t_1,t_2)$ is a diagonal matrix whose diagonal elements are obtained by column stacking of an image chip of size $T_1 \times T_2$ of $\hat{\sigma}_\ell^2(t_1,t_2)$ centered around the pixel (t_1,t_2) in the ℓ-th layer. T_h is the threshold and its value selection is a tradeoff between the probability of detection and the probability of false alarm. Increasing T_h decreases both probability of detection and false alarm. The anomaly detection algorithm is summarized in Table 7.1.

7.4 Performance Evaluation and Simulation Results

In this section, we evaluate the performance of our parameter estimation method and anomaly detection algorithm using simulations.

In the first simulation, we evaluate the performance of the parameter estimation method. We use a two dimensional noncausal ARCH model with the following vector of parameters

$$\boldsymbol{a} = \begin{bmatrix} 1.00 & 0.05 & 0.01 & 0.07 & 0.04 \end{bmatrix}.$$

Table 7.1: Anomaly Detection Algorithm Using Noncausal AR-ARCH model

(1) Transform the image to the wavelet domain and find the feature images using (7.37) through (7.39).
(2) Find the anomaly subspace $\boldsymbol{H}_\ell$ for each layer.
(3) For each layer estimate the AR parameters and the residuals (i.e. $\hat{x}_\ell(t_1,t_2)$) using (7.43).
(4) For each layer estimate the ARCH parameters and the conditional variance (i.e. $\hat{\sigma}_\ell(t_1,t_2)$) using (7.44).
(5) Find the generalized likelihood ratio using (7.49).
(6) Compare the generalized likelihood ratio to a threshold.

$\varepsilon(t_1,t_2)$'s are zero mean independent identically random variables uniformly distributed on the interval $[-\sqrt{3},\sqrt{3}]$. The noncausal ARCH process is obtained using the Banach's fixed point theorem with 15 iterations. The probability density function of the estimated parameters $\widehat{\boldsymbol{a}}_j$ (an estimate of the j-th element of $\boldsymbol{a}$), obtained by 250 Monte-Carlo iterations for different image sizes, are depicted in Figures 7.3 (a) through 7.3 (e). The normalized root mean square error (NRMSE) in the estimation of the parameters is depicted Figure 7.3 (f). Figure 7.3 (f) shows that the mean square error (MSE) in parameter estimation of the proposed estimator converges to zero as the number of data increases which was predicted theoretically in previous section. From Figures 7.3 (a) through 7.3 (e) it can be seen that the estimated parameters converges to Gaussian distributions as as the number of data increases. This phenomenon was proved in [20] and [87] for causal one and multi-dimensional ARCH process, respectively. We conjecture that this is also true for noncausal case, where proving this needs further research.

In the second simulation, we compare the performance of the proposed detection algorithm using synthetic data. We have generated 200 noncausal ARCH(1,1) process with the following vector of parameters

$$\boldsymbol{a} = \begin{bmatrix} 1.00 & 0.03 & 0.04 & 0.05 & 0.06 \end{bmatrix},$$

using the Banach's fixed point theorem with 15 iterations. $\varepsilon(t_1,t_2)$'s were zero mean independent identically random variables uniformly distributed on the interval $[-\sqrt{3},\sqrt{3}]$. The anomaly was chosen to be a 5×5 image with changing location in each iteration. The anomaly subspace

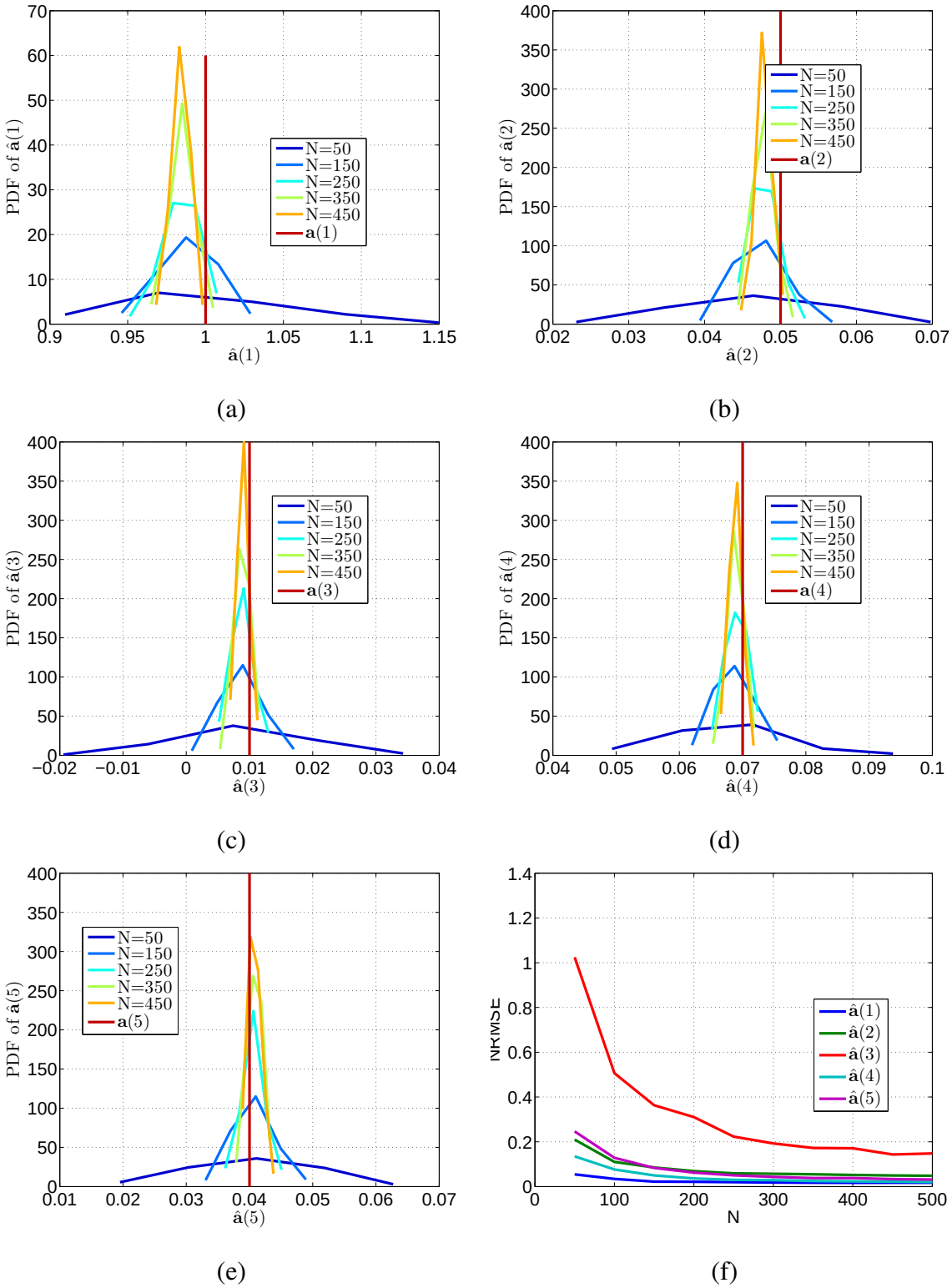


Figure 7.3: PDF of (a): $\hat{\boldsymbol{a}}_1$, (b): $\hat{\boldsymbol{a}}_2$, (c): $\hat{\boldsymbol{a}}_3$, (d): $\hat{\boldsymbol{a}}_4$ and (e): $\hat{\boldsymbol{a}}_5$ for different value of N for an $N \times N$ noncausal ARCH$(1,1)$ process. (f) Estimated normalized root mean square error in parameter estimation of an $N \times N$ noncausal ARCH$(1,1)$ process as a function of N.

$\boldsymbol{H}$ was chosen to be $\boldsymbol{H} = \text{span}\left(\{\boldsymbol{H}_i\}_{i=1}^{4}\right)$ where each $\boldsymbol{H}_i$ was a 5×5 matrix consisting of IID zero mean Gaussian random variables with unity variance. The vector $\boldsymbol{\psi}$ locating the anomaly within anomaly subspace, was chosen randomly such that each of its elements is distributed normally having zero mean and variance 0.5. We compared the performance of anomaly detection using noncausal ARCH modeling with that of GARCH modeling. The parameters of the noncausal ARCH model were estimated via the proposed TSLS method and the parameters of the GARCH model was estimated using maximum likelihood method. To check the sensitivity of the proposed method to anomalies, we estimated the parameters using both images with and without anomaly. We computed the probability of false alarm (P_{fa}) and the probability of detection (P_d) using 200 Monte-Carlo iterations. The receiver operating characteristic (ROC) curves, are depicted in Figure 7.4, where in this figure by theoretic we mean using the image without anomaly to estimate the parameters. As is expected, although the performance of detection is decreased if anomalies present, the performance degradation due to anomalies is not very serious in proposed TSLS method while it is more critical for ML method. It is also obvious that our presented noncausal ARCH model have higher performance comparing to GARCH modeling.

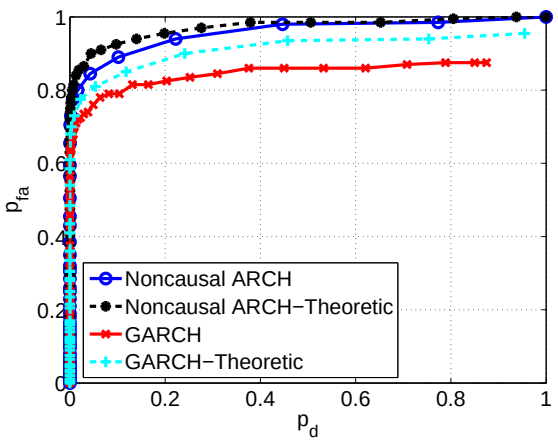


Figure 7.4: ROC curves: comparison of detection performance between noncausal ARCH and GARCH modeling for synthetic data.

In the third simulation, we evaluate the performance of our anomaly detection algorithm. We use real side-scan sonar images. The side-scan sonar images presented in this simulation are from the sonar-5 database collected by the Naval Surface Warfare Center Coastal System

Station (Panama City, FL). The images are 8-bit gray scale. Elongated sea mines (such as those presented in Figure 7.5) are characterized by a bright line (the highlight or echo), corresponding to the scattering effect of mines to the acoustic insonification, and shadow behind them, corresponding to the blocking of sonar waves by mines. Further technical and navigational information about the specific database used, is not available. Since the number of mines in the our database was limited, it was impossible to estimate the anomaly subspace accurately. Instead we used the images in Figure 7.5 to make artificial mines by combining these image linearly using random weights. More specifically, we constructed 100 images by linearly combining these images where the weights are chosen from uniform distribution. The anomaly subspace in each layer (i.e. $\boldsymbol{H}_\ell$) are then estimate from these 100 training images using PCA where we used 10 principle components. Test images where constructed from a real mine-free background together with artificially added mines in random locations. An example of such test image is depicted in Figure 7.6(a) together with true detection which is depicted in Figure 7.6(b). The likelihood ratio (i.e. L_{GLR}) obtained using noncausal AR-ARCH and AR-GARCH model are represented in Figures 7.6(c) and (d), respectively. The vertical and horizontal patch sizes (i.e. $T1$ and T_2) are both set to 16. The wavelet depth, L is set to 3. We used two dimensional undecimated Haar wavelet transform. Other simulation results show that the wavelet basis has no significant effect on the detection results. We used the AR-ARCH model with the following orders : $p = 1, q = 1, r = 2, s = 2$. We compare the performance of our method to that of the method presented in [91]. The ROC curves of both method is depicted in Figure 7.7. These curves are obtained using 100 simulations. As it can be seen from these curves the proposed method has a slightly better performance compared with the method presented in [91]. As mentioned before, the advantage of the proposed method is its lower computational load compared to that of the method proposed in [91].

7.5 Conclusions

We have used the wavelet transform along with a noncausal AR-ARCH model for anomaly detection in side scan sonar images. We introduced a novel background model (i.e. noncausal AR-ARCH model) and used matched subspace detector (MSD) to detect an anomaly in sonar images. We introduced the two dimensional noncausal ARCH model and obtained sufficient

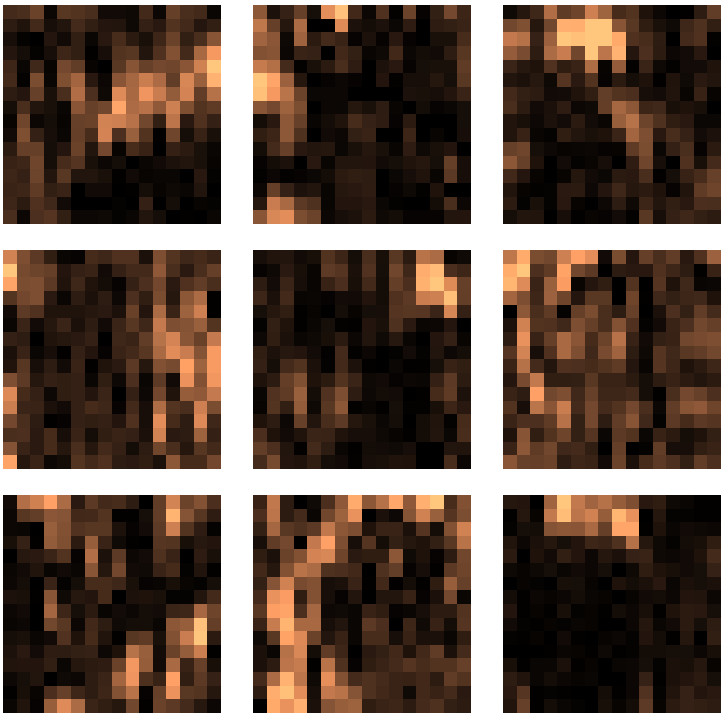
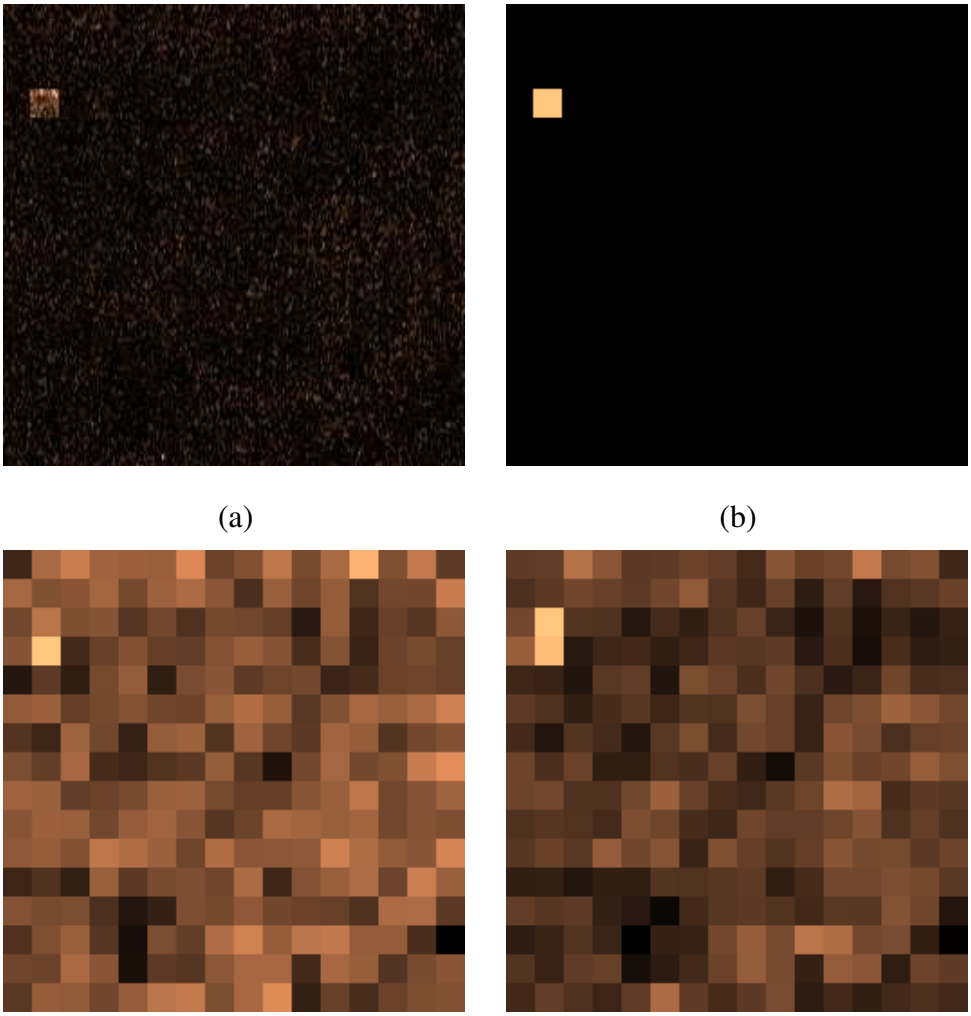


Figure 7.5: Sample side scan sonar image with a sea mine (sea mines are characterized by a bright line (the highlight or echo), corresponding to the scattering effect of the mine to the acoustic insonification, and a shadow behind it, corresponding to the blocking of sonar waves by the mine). These images each 16×16 pixel, are used as training image for obtaining the anomaly subspace.

stationarity conditions. We also proposed the TSLS estimator for parameter estimation of this model. This estimator has a closed form obtained by solving two sets of linear equations. We have shown that this estimator is asymptotically consistent. The most important advantage of this parameter estimation method is its low computational complexity. We also presented an anomaly detection algorithm based on noncausal AR-ARCH modeling of the background. The detection algorithm utilizes a matched subspace detector [101] for the background noncausal AR-ARCH model. This model is an extension of the GARCH model used in [91] for anomaly detection. The advantages of the proposed method over that presented in [91] are lower computational load and lower dependency on the orientation of the image. Simulation results have demonstrated the performance of the proposed method.



(a) (b)

(c) (d)

Figure 7.6: A sample side-scan sonar image with artificial mine (a), true detection (b), generalized likelihood ratio L_{GLR} obtained by noncausal ARCH modeling (c) and GARCH modeling (d.

7.6 Appendix

In this appendix, we show that both proposed estimators i.e. $\hat{\boldsymbol{a}}_{pr}$ and $\hat{\boldsymbol{a}}$ are asymptotically unbiased and consistent estimators. In this section, without loss of generality we assume N and M, (i.e. the number of available data in horizontal and vertical directions, respectively,) are equal. For the sake of simplicity, we change the notation as follows. Instead of double indexing the processes and double sum, we use single indexing and sum. More specifically, by t we mean (t_1,t_2) (hence $y(t) \triangleq y(t_1,t_2)$) and by $\sum_{t=1}^{N^2}$ we mean $\sum_{t_1=1}^{N}\sum_{t_2=1}^{N}$. We assume that the noncausal ARCH process satisfies the conditions (7.3)-(7.5), hence is ergodic. Besides ergodicity, we also

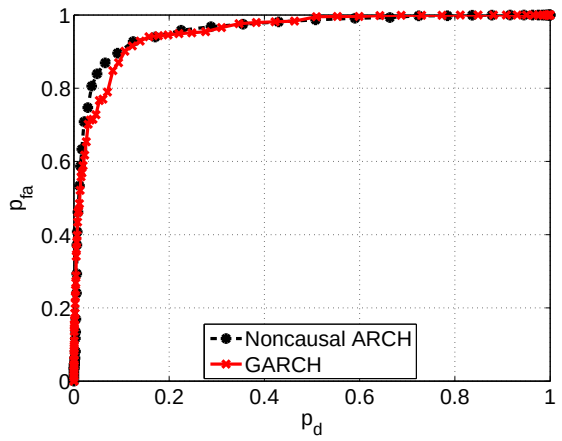


Figure 7.7: ROC curves: comparison of detection performance between noncausal ARCH and GARCH modeling for real data.

assume that

$$\mathbb{E}\left\{\prod_{i=1}^{j} y(t_i)\right\} < \infty \qquad \forall t_i; j = 2,3,4$$
$$\mathbb{E}\left\{\eta_L(t)^2\right\} < \infty \qquad \forall t$$
$$\mathbb{E}\left\{\eta_U(t)^2\right\} < \infty \qquad \forall t \tag{7.52}$$

and $\mathbb{E}\left\{\boldsymbol{y}_L(1)\boldsymbol{y}_L^T(1)\right\}$ and $\mathbb{E}\left\{\boldsymbol{y}_U(1)\boldsymbol{y}_U^T(1)\right\}$ are positive definite matrices having finite elements where $\mathbb{E}\{\cdot\}$ is expectation.

In order to obtain the statistical properties of $\hat{\boldsymbol{a}}_{pr}$, note that

$$\hat{\boldsymbol{a}}_{pr}^T = \frac{1}{2}\left(\boldsymbol{x}_L^T\boldsymbol{R}_L^{-1} + \boldsymbol{x}_U^T\boldsymbol{R}_U^{-1}\right). \tag{7.53}$$

Using the equations (7.14) and (7.15) and the definitions of $\boldsymbol{a}$, $\boldsymbol{y}_L$ and $\boldsymbol{y}_U$, it can be easily verified that

$$y(t) = 2\boldsymbol{a}^T\boldsymbol{y}_L(t) + 2\sigma^2(t)\eta_L(t), \tag{7.54}$$
$$y(t) = 2\boldsymbol{a}^T\boldsymbol{y}_U(t) + 2\sigma^2(t)\eta_U(t). \tag{7.55}$$

Substituting (7.22)-(7.25) into (7.53), we have

$$\begin{aligned}\hat{\boldsymbol{a}}_{pr}^T &= \frac{1}{2}\left(\boldsymbol{x}_L^T\boldsymbol{R}_L^{-1} + \boldsymbol{x}_U^T\boldsymbol{R}_U^{-1}\right)\\ &= \frac{1}{2}\left[\left(\sum_{t=1}^{N^2} y(t)\boldsymbol{y}_L^T(t)\right)\left(2\sum_{t=1}^{N^2}\boldsymbol{y}_L(t)\boldsymbol{y}_L^T(t)\right)^{-1} + \left(\sum_{t=1}^{N^2} y(t)\boldsymbol{y}_U^T(t)\right)\left(2\sum_{t=1}^{N^2}\boldsymbol{y}_U(t)\boldsymbol{y}_U^T(t)\right)^{-1}\right].\end{aligned} \tag{7.56}$$

Substituting (7.54)-(7.55) into (7.56), we have

$$\begin{aligned}\hat{\boldsymbol{a}}_{pr}^T = \frac{1}{2}\Bigg[&\left(\sum_{t=1}^{N^2}\left(2\boldsymbol{a}^T\boldsymbol{y}_L(t)+2\sigma^2(t)\eta_L(t)\right)\boldsymbol{y}_L^T(t)\right)\left(2\sum_{t=1}^{N^2}\boldsymbol{y}_L(t)\boldsymbol{y}_L^T(t)\right)^{-1}+\\&\left(\sum_{t=1}^{N^2}\left(2\boldsymbol{a}^T\boldsymbol{y}_U(t)+2\sigma^2(t)\eta_U(t)\right)\boldsymbol{y}_U^T(t)\right)\left(2\sum_{t=1}^{N^2}\boldsymbol{y}_U(t)\boldsymbol{y}_U^T(t)\right)^{-1}\Bigg]\\=\boldsymbol{a}^T+\frac{1}{2}\Bigg[&\left(\frac{1}{N^2}\sum_{t=1}^{N^2}\sigma^2(t)\eta_L(t)\boldsymbol{y}_L^T(t)\right)\left(\frac{1}{N^2}\sum_{t=1}^{N^2}\boldsymbol{y}_L(t)\boldsymbol{y}_L^T(t)\right)^{-1}+\\&\left(\frac{1}{N^2}\sum_{t=1}^{N^2}\sigma^2(t)\eta_U(t)\boldsymbol{y}_U^T(t)\right)\left(\frac{1}{N^2}\sum_{t=1}^{N^2}\boldsymbol{y}_U(t)\boldsymbol{y}_U^T(t)\right)^{-1}\Bigg].\end{aligned} \tag{7.57}$$

Since we assumed that the process is ergodic, using the weak low of large numbers (WLLN) we have

$$\frac{1}{N^2}\sum_{t=1}^{N^2}\boldsymbol{y}_L(t)\boldsymbol{y}_L^T(t) \xrightarrow{p} \mathbb{E}\left\{\boldsymbol{y}_L(\tau)\boldsymbol{y}_L^T(\tau)\right\}, \tag{7.58}$$

$$\frac{1}{N^2}\sum_{t=1}^{N^2}\boldsymbol{y}_U(t)\boldsymbol{y}_U^T(t) \xrightarrow{p} \mathbb{E}\left\{\boldsymbol{y}_U(\tau)\boldsymbol{y}_U^T(\tau)\right\}, \tag{7.59}$$

where $\tau = [1,1]$. Since $\sigma^2(t) = \boldsymbol{a}^T(\boldsymbol{y}_L(t)+\boldsymbol{y}_U(t))$, we have

$$\begin{aligned}\frac{1}{N^2}\sum_{t=1}^{N^2}\sigma^2(t)\eta_L(t)\boldsymbol{y}_L^T(t) &\xrightarrow{p} \mathbb{E}\left\{\sigma^2(\tau)\eta_L(\tau)\boldsymbol{y}_L^T(\tau)\right\}\\&=\mathbb{E}\left\{\boldsymbol{a}^T(\boldsymbol{y}_L(\tau)+\boldsymbol{y}_U(\tau))\eta_L(\tau)\boldsymbol{y}_L^T(\tau)\right\}\\&=\mathbb{E}\left\{\boldsymbol{a}^T\boldsymbol{y}_L(\tau)\eta_L(\tau)\boldsymbol{y}_L^T(\tau)\right\}+\mathbb{E}\left\{\boldsymbol{a}^T\boldsymbol{y}_U(\tau)\eta_L(\tau)\boldsymbol{y}_L^T(\tau)\right\}\\&=\mathbb{E}\left\{\boldsymbol{a}^T\boldsymbol{y}_L(\tau)\boldsymbol{y}_L^T(\tau)\right\}\mathbb{E}\left\{\eta_L(\tau)\right\}+\mathbb{E}\left\{\boldsymbol{a}^T\boldsymbol{y}_U(\tau)\boldsymbol{y}_L^T(\tau)\right\}\mathbb{E}\left\{\eta_L(\tau)\right\}=\boldsymbol{0},\end{aligned} \tag{7.60}$$

where have used and that fact that $\mathbb{E}\{\eta_L(1)\} = \mathbb{E}\{\eta_U(1)\} = 0$ and $\eta_L(t)$ is uncorrelated with $\boldsymbol{y}_U(\tau)$ and $\boldsymbol{y}_U(\tau)$, as discussed before. Similarly, it can be shown that

$$\frac{1}{N^2}\sum_{t=1}^{N^2}\sigma^2(t)\eta_U(t)\boldsymbol{y}_U^T(t) \xrightarrow{p} \mathbb{E}\left\{\sigma^2(\tau)\eta_U(\tau)\boldsymbol{y}_U^T(\tau)\right\} = \boldsymbol{0}. \tag{7.61}$$

Combining equations (7.57)-(7.61) it follows that

$$\lim_{N\to\infty}\hat{\boldsymbol{a}}_{pr} - \boldsymbol{a} = \boldsymbol{0}, \tag{7.62}$$

and hence

$$\lim_{N\to\infty} \mathbb{E}\left\{\hat{\boldsymbol{a}}_{pr} - \boldsymbol{a}\right\} = \mathbf{0}, \tag{7.63}$$

which shows that $\hat{\boldsymbol{a}}_{pr}$ is an asymptotically unbiased estimator. In order to show that $\hat{\boldsymbol{a}}_{pr}$ is an asymptotically consistent estimator, it suffices to show that

$$\left(\frac{1}{N^2}\sum_{t=1}^{N^2}\sigma^2(t)\eta_L(t)\boldsymbol{y}_L(t)\right)\left(\frac{1}{N^2}\sum_{t=1}^{N^2}\sigma^2(t)\eta_L(t)\boldsymbol{y}_L^T(t)\right) \xrightarrow{p} \mathbf{0} \tag{7.64}$$

$$\left(\frac{1}{N^2}\sum_{t=1}^{N^2}\sigma^2(t)\eta_U(t)\boldsymbol{y}_U(t)\right)\left(\frac{1}{N^2}\sum_{t=1}^{N^2}\sigma^2(t)\eta_U(t)\boldsymbol{y}_U^T(t)\right) \xrightarrow{p} \mathbf{0} \tag{7.65}$$

$$\left(\frac{1}{N^2}\sum_{t=1}^{N^2}\sigma^2(t)\eta_L(t)\boldsymbol{y}_L(t)\right)\left(\frac{1}{N^2}\sum_{t=1}^{N^2}\sigma^2(t)\eta_U(t)\boldsymbol{y}_U^T(t)\right) \xrightarrow{p} \mathbf{0}. \tag{7.66}$$

In order show (7.64), note that

$$\begin{aligned}
&\left(\frac{1}{N^2}\sum_{t=1}^{N^2}\sigma^2(t)\eta_L(t)\boldsymbol{y}_L(t)\right)\left(\frac{1}{N^2}\sum_{t=1}^{N^2}\sigma^2(t)\eta_L(t)\boldsymbol{y}_L^T(t)\right)\\
&= \frac{1}{N^4}\sum_{t_a=1}^{N^2}\sum_{t_b=1}^{N^2}\sigma^2(t_a)\sigma^2(t_b)\eta_L(t_a)\eta_L(t_b)\boldsymbol{y}_L(t_a)\boldsymbol{y}_L^T(t_b)\\
&= \frac{1}{N^4}\left(\sum_{t=1}^{N^2}\sigma^4(t)\eta_L^2(t)\boldsymbol{y}_L(t)\boldsymbol{y}_L^T(t) + \sum_{t_a=1}^{N^2}\sum_{t_b=1,t_b\neq t_a}^{N^2}\sigma^2(t_a)\sigma^2(t_b)\eta_L(t_a)\eta_L(t_b)\boldsymbol{y}_L(t_a)\boldsymbol{y}_L^T(t_b)\right)\\
&= \frac{1}{N^4}\sum_{t=1}^{N^2}(\boldsymbol{a}^T(\boldsymbol{y}_L(t)+\boldsymbol{y}_U(t)))^2\eta_L^2(t)\boldsymbol{y}_L(t)\boldsymbol{y}_L^T(t)\\
&+ \frac{1}{N^4}\sum_{t_a=1}^{N^2}\sum_{t_b=1,t_b\neq t_a}^{N^2}\sigma^2(t_a)\sigma^2(t_b)\eta_L(t_a)\eta_L(t_b)\boldsymbol{y}_L(t_a)\boldsymbol{y}_L^T(t_b).
\end{aligned} \tag{7.67}$$

In order to show that the first term in equation (7.67) converges to zero in probability, note that because of WLLN

$$\frac{1}{N^2}\sum_{t=1}^{N^2}(\boldsymbol{a}^T(\boldsymbol{y}_L(t)+\boldsymbol{y}_U(t)))^2\eta_L^2(t)\boldsymbol{y}_L(t)\boldsymbol{y}_L^T(t) \xrightarrow{p} \mathbb{E}\left\{(\boldsymbol{a}^T(\boldsymbol{y}_L(\tau)+\boldsymbol{y}_U(\tau)))^2\eta_L^2(\tau)\boldsymbol{y}_L(\tau)\boldsymbol{y}_L^T(\tau)\right\} < \infty, \tag{7.68}$$

and is finite since we assumed the process $y(t)$ satisfies conditions (7.52). Hence, the first term in equation (7.67) converges to zero in probability. To show that the second term in equation

(7.67) converges to zero in probability, note that

$$\frac{1}{N^4}\sum_{t_a=1}^{N^2}\sum_{t_b=1,t_b\neq t_a}^{N^2}\sigma^2(t_a)\eta_L(t_a)\eta_L(t_b)\boldsymbol{y}_L(t_a)\boldsymbol{y}_L^T(t_b) = \frac{1}{N^2}\sum_{t_a=1}^{N^2}\sigma^2(t_a)\eta_L(t_a)\boldsymbol{y}_L(t_a)\frac{1}{N^2}\sum_{t_b=1,t_b\neq t_a}^{N^2}\sigma^2(t_b)\eta_L(t_b)\boldsymbol{y}_L^T(t_b), \tag{7.69}$$

where both sum can be shown to converges to zero using the same technique resulted in (7.60). Similarly, it can be easily verified that (7.65) and (7.66) hold, hence $\hat{\boldsymbol{a}}_{pr}$ is an asymptotically consistent estimate of the parameters.

To obtain the statistical properties of $\hat{\boldsymbol{a}}$ note that

$$\hat{\boldsymbol{a}}^T = \frac{1}{2}\left(\overline{\boldsymbol{x}}_L^T\overline{\boldsymbol{R}_L}^{-1} + \overline{\boldsymbol{x}}_U^T\overline{\boldsymbol{R}_L}^{-1}\right). \tag{7.70}$$

Substituting (7.33)-(7.36) into (7.70) and using (7.29)-(7.31), we have

$$\hat{\boldsymbol{a}}^T = \frac{1}{2}\left[\left(\sum_{t=1}^{N^2}\frac{y(t)\boldsymbol{y}_L^T(t)}{\hat{\sigma}^4(t)}\right)\left(2\sum_{t=1}^{N^2}\frac{\boldsymbol{y}_L(t)\boldsymbol{y}_L^T(t)}{\hat{\sigma}^4(t)}\right)^{-1} + \left(\sum_{t=1}^{N^2}\frac{y(t)\boldsymbol{y}_U^T(t)}{\hat{\sigma}^4(t)}\right)\left(2\sum_{t=1}^{N^2}\frac{\boldsymbol{y}_U(t)\boldsymbol{y}_U^T(t)}{\hat{\sigma}^4(t)}\right)^{-1}\right]. \tag{7.71}$$

Inserting (7.54)-(7.55) into (7.71), we have

$$\begin{aligned}\hat{\boldsymbol{a}}^T &= \frac{1}{2}\left[\left(\sum_{t=1}^{N^2}\frac{\left(\boldsymbol{a}^T\boldsymbol{y}_L(t)+\sigma^2(t)\eta_L(t)\right)\boldsymbol{y}_L^T(t)}{\hat{\sigma}^4(t)}\right)\left(\sum_{t=1}^{N^2}\frac{\boldsymbol{y}_L(t)\boldsymbol{y}_L^T(t)}{\hat{\sigma}^4(t)}\right)^{-1}\right.\\ &\quad\left.+\left(\sum_{t=1}^{N^2}\frac{\left(\boldsymbol{a}^T\boldsymbol{y}_U(t)+\sigma^2(t)\eta_U(t)\right)\boldsymbol{y}_U^T(t)}{\hat{\sigma}^4(t)}\right)\left(\sum_{t=1}^{N^2}\frac{\boldsymbol{y}_U(t)\boldsymbol{y}_U^T(t)}{\hat{\sigma}^4(t)}\right)^{-1}\right]\\ &= \boldsymbol{a}^T + \frac{1}{2}\left[\left(\sum_{t=1}^{N^2}\frac{\sigma^2(t)\eta_L(t)\boldsymbol{y}_L^T(t)}{\hat{\sigma}^4(t)}\right)\left(\sum_{t=1}^{N^2}\frac{\boldsymbol{y}_L(t)\boldsymbol{y}_L^T(t)}{\hat{\sigma}^4(t)}\right)^{-1}\right.\\ &\quad\left.+\left(\sum_{t=1}^{N^2}\frac{\sigma^2(t)\eta_U(t)\boldsymbol{y}_U^T(t)}{\hat{\sigma}^4(t)}\right)\left(\sum_{t=1}^{N^2}\frac{\boldsymbol{y}_U(t)\boldsymbol{y}_U^T(t)}{\hat{\sigma}^4(t)}\right)^{-1}\right].\end{aligned} \tag{7.72}$$

Using a very similar approach as used in [87] it can be shown that, in (7.72) we can substitute $\hat{\sigma}^2(t)$ with $\sigma^2(t)$ and we get

$$\begin{aligned}\hat{\boldsymbol{a}}^T - \boldsymbol{a}^T &= \frac{1}{2}\left[\left(\sum_{t=1}^{N^2}\frac{\eta_L(t)\boldsymbol{y}_L^T(t)}{\sigma^2(t)}\right)\left(\sum_{t=1}^{N^2}\frac{\boldsymbol{y}_L(t)\boldsymbol{y}_L^T(t)}{\sigma^2(t)}\right)^{-1}\right.\\ &\quad\left.+\left(\sum_{t=1}^{N^2}\frac{\eta_U(t)\boldsymbol{y}_U^T(t)}{\sigma^2(t)}\right)\left(\sum_{t=1}^{N^2}\frac{\boldsymbol{y}_U(t)\boldsymbol{y}_U^T(t)}{\sigma^2(t)}\right)^{-1}\right] + o_p(1).\end{aligned} \tag{7.73}$$

Since $\sigma^2(t) = \boldsymbol{a}^T(\boldsymbol{y}_L(t) + \boldsymbol{y}_U(t))$ we have

$$\hat{\boldsymbol{a}}^T - \boldsymbol{a}^T = \frac{1}{2}\left[\left(\frac{1}{N^2}\sum_{t=1}^{N^2}\frac{\eta_L(t)\boldsymbol{y}_L^T(t)}{\boldsymbol{a}^T(\boldsymbol{y}_L(t)+\boldsymbol{y}_U(t))}\right)\left(\frac{1}{N^2}\sum_{t=1}^{N^2}\frac{\boldsymbol{y}_L(t)\boldsymbol{y}_L^T(t)}{\boldsymbol{a}^T(\boldsymbol{y}_L(t)+\boldsymbol{y}_U(t))}\right)^{-1} + \left(\frac{1}{N^2}\sum_{t=1}^{N^2}\frac{\eta_U(t)\boldsymbol{y}_U^T(t)}{\boldsymbol{a}^T(\boldsymbol{y}_L(t)+\boldsymbol{y}_U(t))}\right)\left(\frac{1}{N^2}\sum_{t=1}^{N^2}\frac{\boldsymbol{y}_U(t)\boldsymbol{y}_U^T(t)}{\boldsymbol{a}^T(\boldsymbol{y}_L(t)+\boldsymbol{y}_U(t))}\right)^{-1}\right] + o_p(1). \tag{7.74}$$

With ergodicity assumption and use of WLLN, we have

$$\frac{1}{N^2}\sum_{t=1}^{N^2}\frac{\boldsymbol{y}_L(t)\boldsymbol{y}_L^T(t)}{\boldsymbol{a}^T(\boldsymbol{y}_L(t)+\boldsymbol{y}_U(t))} \xrightarrow{p} \mathbb{E}\left\{\frac{\boldsymbol{y}_L(\tau)\boldsymbol{y}_L^T(\tau)}{\boldsymbol{a}^T(\boldsymbol{y}_L(\tau)+\boldsymbol{y}_U(\tau))}\right\}, \tag{7.75}$$

$$\frac{1}{N^2}\sum_{t=1}^{N^2}\frac{\boldsymbol{y}_U(t)\boldsymbol{y}_U^T(t)}{\boldsymbol{a}^T(\boldsymbol{y}_L(t)+\boldsymbol{y}_U(t))} \xrightarrow{p} \mathbb{E}\left\{\frac{\boldsymbol{y}_U(\tau)\boldsymbol{y}_U^T(\tau)}{\boldsymbol{a}^T(\boldsymbol{y}_L(\tau)+\boldsymbol{y}_U(\tau))}\right\}. \tag{7.76}$$

Note that since we assumed that c_0 is absolutely positive, then $\sigma^2(\tau) > 0$ and because of the moment constraints (7.52) we assumed, these quantities are finite. Furthermore, we have

$$\frac{1}{N^2}\sum_{t=1}^{N^2}\frac{\eta_L(t)\boldsymbol{y}_L^T(t)}{\boldsymbol{a}^T(\boldsymbol{y}_L(t)+\boldsymbol{y}_U(t))} \xrightarrow{p} \mathbb{E}\left\{\frac{\eta_L(\tau)\boldsymbol{y}_L^T(\tau)}{\boldsymbol{a}^T(\boldsymbol{y}_L(\tau)+\boldsymbol{y}_U(\tau))}\right\}, \tag{7.77}$$

$$\frac{1}{N^2}\sum_{t=1}^{N^2}\frac{\eta_U(t)\boldsymbol{y}_U^T(t)}{\boldsymbol{a}^T(\boldsymbol{y}_L(t)+\boldsymbol{y}_U(t))} \xrightarrow{p} \mathbb{E}\left\{\frac{\eta_U(\tau)\boldsymbol{y}_U^T(\tau)}{\boldsymbol{a}^T(\boldsymbol{y}_L(\tau)+\boldsymbol{y}_U(\tau))}\right\}. \tag{7.78}$$

Using the fact that $\mathbb{E}\{\eta_L(1)\} = \mathbb{E}\{\eta_U(1)\} = 0$ and $\eta_L(t)$ is uncorrelated with $\boldsymbol{y}_U(\tau)$ and $\boldsymbol{y}_U(\tau)$, we have

$$\mathbb{E}\left\{\frac{\eta_L(\tau)\boldsymbol{y}_L^T(\tau)}{\boldsymbol{a}^T(\boldsymbol{y}_L(\tau)+\boldsymbol{y}_U(\tau))}\right\} = \mathbb{E}\left\{\frac{\boldsymbol{y}_L^T(\tau)}{\boldsymbol{a}^T(\boldsymbol{y}_L(\tau)+\boldsymbol{y}_U(\tau))}\right\}\mathbb{E}\{\eta_L\} = \boldsymbol{0}, \tag{7.79}$$

$$\mathbb{E}\left\{\frac{\eta_U(\tau)\boldsymbol{y}_U^T(\tau)}{\boldsymbol{a}^T(\boldsymbol{y}_L(\tau)+\boldsymbol{y}_U(\tau))}\right\} = \mathbb{E}\left\{\frac{\eta_U(\tau)\boldsymbol{y}_U^T(\tau)}{\boldsymbol{a}^T(\boldsymbol{y}_L(\tau)+\boldsymbol{y}_U(\tau))}\right\}\mathbb{E}\{\eta_U\} = \boldsymbol{0}. \tag{7.80}$$

Combining equations (7.72)-(7.80) it follows that

$$\lim_{N\to\infty} \hat{\boldsymbol{a}} - \boldsymbol{a} = \boldsymbol{0}, \tag{7.81}$$

and hence

$$\lim_{N\to\infty} \mathbb{E}\{\hat{\boldsymbol{a}} - \boldsymbol{a}\} = \boldsymbol{0}, \tag{7.82}$$

which shows that $\hat{\boldsymbol{a}}$ is an asymptotically unbiased estimator.

In order to show that $\hat{\boldsymbol{a}}$ is an asymptotically consistent estimator, it suffices to show that

$$\left(\frac{1}{N^2}\sum_{t=1}^{N^2}\frac{\eta_L(t)\boldsymbol{y}_L(t)}{\boldsymbol{a}^T(\boldsymbol{y}_L(t)+\boldsymbol{y}_U(t))}\right)\left(\frac{1}{N^2}\sum_{t=1}^{N^2}\frac{\eta_L(t)\boldsymbol{y}_L^T(t)}{\boldsymbol{a}^T(\boldsymbol{y}_L(t)+\boldsymbol{y}_U(t))}\right)\xrightarrow{p}\mathbf{0}, \tag{7.83}$$

$$\left(\frac{1}{N^2}\sum_{t=1}^{N^2}\frac{\eta_L(t)\boldsymbol{y}_U(t)}{\boldsymbol{a}^T(\boldsymbol{y}_L(t)+\boldsymbol{y}_U(t))}\right)\left(\frac{1}{N^2}\sum_{t=1}^{N^2}\frac{\eta_L(t)\boldsymbol{y}_U^T(t)}{\boldsymbol{a}^T(\boldsymbol{y}_L(t)+\boldsymbol{y}_U(t))}\right)\xrightarrow{p}\mathbf{0}, \tag{7.84}$$

$$\left(\frac{1}{N^2}\sum_{t=1}^{N^2}\frac{\eta_L(t)\boldsymbol{y}_U(t)}{\boldsymbol{a}^T(\boldsymbol{y}_L(t)+\boldsymbol{y}_U(t))}\right)\left(\frac{1}{N^2}\sum_{t=1}^{N^2}\frac{\eta_L(t)\boldsymbol{y}_L^T(t)}{\boldsymbol{a}^T(\boldsymbol{y}_L(t)+\boldsymbol{y}_U(t))}\right)\xrightarrow{p}\mathbf{0}. \tag{7.85}$$

In order to show (7.83) is correct, note that

$$\begin{aligned}
&\left(\frac{1}{N^2}\sum_{t=1}^{N^2}\frac{\eta_L(t)\boldsymbol{y}_L(t)}{\boldsymbol{a}^T(\boldsymbol{y}_L(t)+\boldsymbol{y}_U(t))}\right)\left(\frac{1}{N^2}\sum_{t=1}^{N^2}\frac{\eta_L(t)\boldsymbol{y}_L^T(t)}{\boldsymbol{a}^T(\boldsymbol{y}_L(t)+\boldsymbol{y}_U(t))}\right)\\
&=\frac{1}{N^4}\sum_{t_a=1}^{N^2}\sum_{t_b=1}^{N^2}\frac{\eta_L(t_a)\boldsymbol{y}_L(t_a)}{\boldsymbol{a}^T(\boldsymbol{y}_L(t_a)+\boldsymbol{y}_U(t_a))}\frac{\eta_L(t_b)\boldsymbol{y}_L^T(t_b)}{\boldsymbol{a}^T(\boldsymbol{y}_L(t_b)+\boldsymbol{y}_U(t_b))}\\
&=\frac{1}{N^4}\sum_{t=1}^{N^2}\frac{\eta_L(t)^2\boldsymbol{y}_L(t)\boldsymbol{y}_L^T(t)}{(\boldsymbol{a}^T(\boldsymbol{y}_L(t)+\boldsymbol{y}_U(t)))^2}+\frac{1}{N^4}\sum_{t_a=1}^{N^2}\sum_{t_b=1,t_b\neq t_a}^{N^2}\frac{\eta_L(t_a)\boldsymbol{y}_L(t_a)}{\boldsymbol{a}^T(\boldsymbol{y}_L(t_a)+\boldsymbol{y}_U(t_a))}\frac{\eta_L(t_b)\boldsymbol{y}_L^T(t_b)}{\boldsymbol{a}^T(\boldsymbol{y}_L(t_b)+\boldsymbol{y}_U(t_b))}.
\end{aligned} \tag{7.86}$$

In order to show that (7.86) converge to zero in probability, note that, for the first term in (7.86) using WLLN we have

$$\frac{1}{N^2}\sum_{t=1}^{N^2}\frac{\eta_L(t)^2\boldsymbol{y}_L(t)\boldsymbol{y}_L^T(t)}{(\boldsymbol{a}^T(\boldsymbol{y}_L(t)+\boldsymbol{y}_U(t)))^2}\xrightarrow{p}\mathbb{E}\left\{\frac{\eta_L(\tau)^2\boldsymbol{y}_L(\tau)\boldsymbol{y}_L^T(\tau)}{(\boldsymbol{a}^T(\boldsymbol{y}_L(\tau)+\boldsymbol{y}_U(\tau)))^2}\right\}, \tag{7.87}$$

which is finite because the moment constraints (i.e. (7.52)) we assumed and the fact that $\left(\boldsymbol{a}^T(\boldsymbol{y}_L(\tau)+\boldsymbol{y}_U(\tau))\right)^2>0$ since $c_0>0$. Hence,

$$\frac{1}{N^4}\sum_{t=1}^{N^2}\frac{\eta_L(t)^2\boldsymbol{y}_L(t)\boldsymbol{y}_L^T(t)}{(\boldsymbol{a}^T(\boldsymbol{y}_L(t)+\boldsymbol{y}_U(t)))^2}\xrightarrow{p}\mathbf{0}. \tag{7.88}$$

In order to show that the second term in (7.86) converges to zero in probability, note that,

$$\begin{aligned}
&\frac{1}{N^4}\sum_{t_a=1}^{N^2}\sum_{t_b=1,t_b\neq t_a}^{N^2}\frac{\eta_L(t_a)\boldsymbol{y}_L(t_a)}{\boldsymbol{a}^T(\boldsymbol{y}_L(t_a)+\boldsymbol{y}_U(t_a))}\frac{\eta_L(t_b)\boldsymbol{y}_L^T(t_b)}{\boldsymbol{a}^T(\boldsymbol{y}_L(t_b)+\boldsymbol{y}_U(t_b))}\\
&=\frac{1}{N^2}\sum_{t_a=1}^{N^2}\left(\frac{\eta_L(t_a)\boldsymbol{y}_L(t_a)}{\boldsymbol{a}^T(\boldsymbol{y}_L(t_a)+\boldsymbol{y}_U(t_a))}\right)\frac{1}{N^2}\sum_{t_b=1,t_b\neq t_a}^{N^2}\left(\frac{\eta_L(t_b)\boldsymbol{y}_L^T(t_b)}{\boldsymbol{a}^T(\boldsymbol{y}_L(t_b)+\boldsymbol{y}_U(t_b))}\right),
\end{aligned} \tag{7.89}$$

and using WLLN and the fact that $\mathbb{E}\{\eta_L(1)\}=0$ and $\eta_L(t)$ is uncorrelated with $\boldsymbol{y}_U(\tau)$ and

$\boldsymbol{y}_U(\tau)$, we have

$$\frac{1}{N^2}\sum_{t_a=1}^{N^2}\left(\frac{\eta_L(t_a)\boldsymbol{y}_L(t_a)}{\boldsymbol{a}^T(\boldsymbol{y}_L(t_a)+\boldsymbol{y}_U(t_a))}\right) \xrightarrow{p} \mathbb{E}\left\{\frac{\eta_L(\tau)\boldsymbol{y}_L(\tau)}{\boldsymbol{a}^T(\boldsymbol{y}_L(\tau)+\boldsymbol{y}_U(\tau))}\right\}$$
$$= \mathbb{E}\left\{\frac{\boldsymbol{y}_L(\tau)}{\boldsymbol{a}^T(\boldsymbol{y}_L(\tau)+\boldsymbol{y}_U(\tau))}\right\}\mathbb{E}\left\{\eta_L(\tau)\right\} = \mathbf{0}. \tag{7.90}$$

Hence (7.86) converges to zero in probability and therefore (7.83) holds. Using the similar approach it can be verified that (7.84)-(7.85) hold, and hence $\hat{\boldsymbol{a}}$ is an asymptotically consistent estimator. This is the main result of this chapter.

Chapter 8

ARCH and GARCH Parameter Estimation in Presence of Additive Noise Using Particle Methods

8.1 Introduction

Parameter estimation is the backbone of all model based signal processing algorithms. The generalized autoregressive conditional heteroscedasticity (GARCH) model was first introduced by Bollerslev [19] as an extension of the autoregressive conditional heteroscedasticity (ARCH) model developed by Engle [42] to model econometric data. Since then, many researchers have extended and used these models in several speech and image processing applications. Cohen [25] modeled the speech signal in the short time Fourier transform (STFT) domain as a complex GARCH process and used this model for speech enhancement. AR-GARCH model was utilized for modeling speech signal in the time domain and for developing voice activity detection (VAD) algorithms [84, 112]. Abdolahi et al. [2] used the parameters of the GARCH model for speech recognition in Persian isolated digits. Noiboar and Cohen [91] used the GARCH model for anomaly detection in sonar images. In all of the above mentioned uses of GARCH models in speech signal processing, the authors assume that the parameters of the GARCH model are known or can be estimated from a database.

Parameter estimation methods for ARCH and GARCH models, such as quasi maximum likelihood (QML) [129], two stage least squares (TSLS) [20], and constrained two stage least

squares (CTSLS) [86], assume that the data are clean (without any additive noise). So in most speech signal processing algorithms these methods are often inapplicable due to presence of additive noise. Recently, Mousazadeh and Cohen [83] proposed an ML method for simultaneous parameter estimation and state smoothing of complex (G)ARCH processes in the presence of additive noise. In that work, it is assumed that the probability density functions (pdfs) of the additive corrupting noise and the process noise (v_k in the definition of GARCH (8.1)) are Gaussian. This assumption restricts the application of the method in general cases. Poyiadjis et al. [95] introduced a method for parameter estimation of general state-space models using particle methods. In their method, they find the ML estimate of the parameters. The drawback of their method is that it does not consider the cases that some constraints such as stationarity are imposed on the model. ARCH and GARCH parameter estimation in presence of noise is one of these cases.

In this chapter, we propose a gradient based technique for estimating the parameters of ARCH and GARCH models in presence of additive noise. More specifically, we use a gradient descent method together with active set method to obtain the maximum of the likelihood function. The gradient of the likelihood function of noisy observations given the parameters which is needed for gradient based optimization algorithm, is obtained by particle method. The reminder of this chapter is as follows. In Section 8.2, we formulate the problem and introduce a novel technique for estimating the parameters of ARCH and GARCH models in the presence of noise considering the stationarity constraints. Simulation results and performance comparison are presented in Section 8.3. We conclude the chapter in Section 8.4.

8.2 Problem Formulation and the Proposed Method for Parameter Estimation

GARCH model of order (q,p) is defined as [19]

$$z_k = \sigma_k v_k \tag{8.1}$$

$$\sigma_k = \left(c_0 + \sum_{i=1}^{q} b_i z_{k-i}^2 + \sum_{i=1}^{p} a_i \sigma_{k-i}^2 \right)^{1/2} \tag{8.2}$$

where $\boldsymbol{\theta} = \left[c_0, b_1, \cdots, b_q, a_1, \cdots, a_p\right]$ is the vector of parameters of the GARCH model, $(\cdot)^T$ is the transpose operator and v_k's are independent identically distributed random variables with zero mean and unit variance. ARCH model is a special case of GARCH model with $a_i = 0, \forall i$. Necessary and sufficient conditions for stationarity with finite second order moment are [19]

$$\begin{aligned} & c_0 > 0 \\ & b_i \geq 0 \qquad ; \qquad i = 1, 2, \cdots, q \\ & a_i \geq 0 \qquad ; \qquad i = 1, 2, \cdots, p \\ & \sum_{i=1}^{q} b_i + \sum_{i=1}^{p} a_i < 1. \end{aligned} \tag{8.3}$$

Now, suppose that the observation signal, y_k, is a GARCH process corrupted with an additive noise, i.e.

$$y_k = z_k + n_k \tag{8.4}$$

where n_k is the additive noise sequence. For the rest of the chapter we assume that the probability density functions of v_k and n_k are completely known except for some unknown parameters such as the noise level. If we define the hidden state vector as

$$\boldsymbol{x}_k = \begin{bmatrix} z_k & \cdots & z_{k-q+1} & \sigma_k & \cdots & \sigma_{k-p+1} \end{bmatrix}^T \tag{8.5}$$

then we will have

$$\boldsymbol{x}_{k+1} = F\left(\boldsymbol{x}_k, v_k, \boldsymbol{\theta}\right) \tag{8.6}$$

$$y_k = G\left(x_k, n_k, \boldsymbol{\theta}\right) \tag{8.7}$$

where

$$F\left(\boldsymbol{x}_k, v_k, \boldsymbol{\theta}\right) = \begin{bmatrix} \sqrt{c_0 + \sum_{i=1}^{q} b_i z_{k-i}^2 + \sum_{i=1}^{p} a_i \sigma_{k-i}^2} v_k \\ z_{k-1} \\ \vdots \\ z_{k-p+1} \\ \sqrt{c_0 + \sum_{i=1}^{q} b_i z_{k-i}^2 + \sum_{i=1}^{p} a_i \sigma_{k-i}^2} \\ \sigma_{k-1} \\ \vdots \\ \sigma_{k-p+1} \end{bmatrix} \tag{8.8}$$

$$G\left(\boldsymbol{x}_k, n_k, \boldsymbol{\theta}\right) = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \boldsymbol{x}_k + n_k. \tag{8.9}$$

Reducing this model to ARCH process corrupted with additive noise is straightforward. It is worth mentioning that if the probability density functions of v_k and n_k have some unknown parameters, we can modify the vector of parameters such that it includes these unknown parameters. Available methods for estimating the parameters in this general state space (i.e. equations) model are extended Kalman Filters (EKF), unscented Kalman filter (UKF) and particle filters (PF)[57]. Poyiadjis et al. [95], introduced a method for parameter estimation in the general state-space model given by (8.8) and (8.9), using particle methods. This algorithm is based on finding the likelihood function of the observations in terms of the vector of parameters θ i.e. $p(y_0, y_1, \cdots, y_{K-1}; \theta)$, using particle methods and finding the maximum of this function by a gradient decent method.

In our case, i.e. parameter estimation for ARCH and GARCH models in presence of additive noise, stationarity and finite second order moment constraints must be considered. Since the method presented in [95] gives an estimate of the gradient vector of the probability density function of the observations, we can use this estimate along with the active set and gradient projection methods [76] to solve our constrained maximization problem. The constraints on parameters which are given by (8.3) can be restated as follows

$$\begin{bmatrix} -\boldsymbol{I}_{p+q+1} \\ \mathbf{1}_{1\times(p+q+1)} \end{bmatrix} \begin{bmatrix} c_0 \\ b_1 \\ \vdots \\ b_q \\ a_1 \\ \vdots \\ a_p \end{bmatrix} = \boldsymbol{A\theta} \leq \begin{bmatrix} \varepsilon \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \tag{8.10}$$

where $\boldsymbol{I}_d$ is $d \times d$ identity matrix, $\mathbf{1}_{n\times m}$ is $n \times m$ matrix with all of its elements equal to one, ε is a very small number greater than zero. By $\boldsymbol{a} \leq \boldsymbol{b}$, we mean that each element in $\boldsymbol{a}$ is smaller than or equal to the corresponding element in $\boldsymbol{b}$. Our algorithm is proposed in Table 8.1. It is worth mentioning that the proposed algorithm can be used in any parameter estimation problem with linear constraints. It is worth mentioning that the proposed method can be modified to obtain a recursive ML algorithm, which can be applied to the case where the parameters are time varying, see [83] for further information.

In the algorithm given in Table 8.1, N is the number of particles, $y_0^{K-1} = \{y_0, y_1, \cdots, y_{K-1}\}$

Table 8.1: Proposed Algorithm for Estimating the Parameters of a Linearly Constrained Model Using Particle Methods

1. Choose an initial value ($\boldsymbol{\theta}_0$) for the parameter vector.
2. Find the set of active constraints and form the matrix $\boldsymbol{A}_c$ by removing the rows corresponding to inactive constraints from matrix $\boldsymbol{A}$.
3. For $k = 0$ to $k = K-1$ do the following
   (a) For $i = 1, 2, \cdots, N$ sample $\chi_k^{(i)} \sim q\left(\chi_k^{(i)} | \boldsymbol{x}_{k-1}, y_0^k; \boldsymbol{\theta}\right) = \sum_{i=1}^N \widetilde{a}_{k-1}^{(i)} q_{\boldsymbol{\theta}_k}\left(\cdot | y_0^k, \chi_{k-1}^{(i)}\right)$.
   (b) Find the values of $a_k^{(i)}$ and $\rho_k^{(i)}$ using
   $$a_k^{(i)} = \frac{\hat{\zeta}_{\boldsymbol{\theta}}\left(y_0^k, \chi_k^{(i)}\right)}{q(\chi_k^{(i)} | \boldsymbol{x}_{k-1}, y_0^k; \boldsymbol{\theta})},$$
   $$\rho_k^{(i)} = \frac{\nabla\zeta(y_0^k, \chi_k^{(i)})}{q(\chi_k^{(i)} | \boldsymbol{x}_{k-1}, y_0^k; \boldsymbol{\theta})}.$$
   (c) Find the values of $\widetilde{a}_k^{(i)}$ and $\widetilde{\beta}_k^{(i)}$ using
   $$\widetilde{a}_k^{(i)} = \frac{a_k^{(i)}}{\sum_{i=1}^N a_k^{(i)}}$$
   $$\widetilde{a}_k^{(i)} \widetilde{\beta}_k^{(i)} = \frac{\rho_k^{(i)}}{\sum_{i=1}^N a_k^{(i)}} + \widetilde{a}_k^{(i)} \frac{\sum_{i=1}^N \rho_k^{(i)}}{\sum_{i=1}^N a_k^{(i)}}$$
4. Calculate the value of gradient of the pdf using $\boldsymbol{g} = \nabla \log\left(\hat{p}(y_0^{K-1}; \boldsymbol{\theta})\right) = \sum_{k=0}^{K-1} \frac{\sum_{i=1}^N \rho_k^{(i)}}{\sum_{i=1}^N a_k^{(i)}}$.
5. Calculate the following quantities
   $$\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{A}_c^T (\boldsymbol{A}_c \boldsymbol{A}_c^T)^{-1} \boldsymbol{A}_c,$$
   $$\boldsymbol{d} = \boldsymbol{P}\boldsymbol{g}^T.$$
6. If $\boldsymbol{d} = 0$ then find α_1 and α_2 such that $\alpha_1 = \max\{\alpha | \boldsymbol{\theta} + \alpha d \text{ is feasible }, \alpha \in \mathrm{R}^+\}$.
   Select a value for α_2 such that $0 \leq \alpha_2 \leq \alpha_1$. Set $\boldsymbol{\theta} = \boldsymbol{\theta} + \alpha_2 \boldsymbol{d}$ and go to 2.
7. If $\boldsymbol{d} \neq 0$, compute
   $\boldsymbol{\lambda} = -\boldsymbol{A}_c^T (\boldsymbol{A}_c \boldsymbol{A}_c^T)^{-1} \boldsymbol{A}_c \boldsymbol{g}$ and do the following steps
   (a) If for all j corresponding to active constraints the j-th element of $\boldsymbol{\lambda}$ is greater or equal to zero, then $\boldsymbol{\theta}$ is optimum, and the algorithm is finished.
   (b) Else, eliminate the row corresponding to the most negative element of $\boldsymbol{\lambda}$ from the matrix $\boldsymbol{A}_c$ and go to 3.

and $\widetilde{\zeta}_{\boldsymbol{\theta}}(y_0^k, \chi_k^{(i)})$ and $\widetilde{\nabla\zeta}_{\boldsymbol{\theta}}(y_0^k, \chi_k^{(i)})$ are computed by the following equations.

$$\widetilde{\zeta}_{\boldsymbol{\theta}}(y_0^k, \chi_k^{(i)}) = \sum_{i=1}^{N} \widetilde{a}_{k-1}^{(i)} g_{\boldsymbol{\theta}}(y_k|\chi_k^{(i)}) f_{\boldsymbol{\theta}}(\chi_k^{(i)}|\chi_{k-1}^{(i)}) \tag{8.11}$$

$$\widetilde{\nabla\zeta}_{\boldsymbol{\theta}}(y_0^k, \chi_k^{(i)}) = \sum_{i=1}^{N} \widetilde{a}_{k-1}^{(i)} g_{\boldsymbol{\theta}}(y_k|\chi_k^{(i)}) f_{\boldsymbol{\theta}}(\chi_k^{(i)}|\chi_{k-1}^{(i)}) \times \left[\nabla\left(\log g_{\boldsymbol{\theta}}(y_k|\chi_k^{(i)})\right) + \nabla \log\left(f_{\boldsymbol{\theta}}(\chi_k^{(i)}|\chi_{k-1}^{(i)})\right) + \widetilde{\beta}_{k-1}^{(i)}\right] \tag{8.12}$$

where $g_{\boldsymbol{\theta}}(y_k|\chi_k^{(i)})$ and $f_{\boldsymbol{\theta}}(\chi_k^{(i)}|\chi_{k-1}^{(i)})$ are conditional transition and observation distributions, respectively, i.e.,

$$y_k|\boldsymbol{x}_k \sim g_{\boldsymbol{\theta}}(y_k|\boldsymbol{x}_k) \tag{8.13}$$

$$\boldsymbol{x}_k|\boldsymbol{x}_{k-1} \sim f_{\boldsymbol{\theta}}(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \tag{8.14}$$

In this algorithm $q_{\boldsymbol{\theta}_k}(.|y_0^k, \chi_{k-1}^{(i)})$ is the assumed distribution and is case dependent. The best choice of this distribution is $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, y_0^k; \boldsymbol{\theta})$ which is the conditional distribution of the current state, $\boldsymbol{x}_k$, conditioned on the previous state and y_0^k. In most problems, computation of this distribution and generating samples from it is almost impossible. So, the usual choice for the distribution is the transition distribution, i.e.,

$$q_{\boldsymbol{\theta}_k}(\cdot|y_0^k, \chi_{k-1}^{(i)}) = f_{\boldsymbol{\theta}_k}(\cdot|\chi_{k-1}^{(i)}). \tag{8.15}$$

This algorithm can be easily applied to ARCH and GARCH parameter estimation problems in presence of additive noise. The main point is obtaining the conditional observation and transition distributions as follows

$$g_{\boldsymbol{\theta}}(y_k|\boldsymbol{x}_k) = p_n(y_k - \boldsymbol{x}_k(1)) \tag{8.16}$$

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) = \frac{1}{\boldsymbol{x}_k(q+1)} p_v\left(\frac{x_k(1)}{\boldsymbol{x}_k(q+1)}\right) \times \prod_{i=2, i\neq q+1}^{p+q} \delta(\boldsymbol{x}_k(i) - \boldsymbol{x}_{k-1}(i-1)) \times \delta\left(\boldsymbol{x}_k(q+1) - \sqrt{c_0 + \sum_{i=1}^{q} b_i \boldsymbol{x}_{k-1}^2(i) + \sum_{i=1}^{p} a_i \boldsymbol{x}_{k-1}^2(i+q)}\right) \tag{8.17}$$

where $\boldsymbol{x}_k(j)$ is the j-th element of $\boldsymbol{x}_k$, $\delta(\cdot)$ is the Dirac delta function and p_v and p_n are the distributions of v_k and n_k, respectively.

8.3 simulation results

In this section, we investigate the performance of the proposed method PCon (the algorithm depicted in Table 8.1) compared with four ML methods. The first one, denoted by MLC, employs the clean data (unavailable in practical situations) for estimating the parameters.The second one, denoted by MLN, employs the noisy data for estimating the parameters assuming that the data are clean. The third one, denoted by MLG, is the one proposed in [83], assuming both the process noise and corrupting noise are Gaussian. The last method, denoted by PUnc (the algorithm presented in [95]) employs particle methods without considering the stationarity constraints.

Table 8.2: NRMSE in Parameter Estimation For Different Methods

NMSE	MLC	MLN	MLG	PUnc	PCon
c_0	0.2436	4.2774	0.4688	0.3945	0.2948
a_1	0.2116	0.6594	0.4718	0.4175	0.2475
b_1	0.2991	0.7414	0.6758	0.5432	0.3864

For this simulation, we used a GARCH$(1,1)$ process with Laplace process noise, corrupted by a zero mean Gaussian white noise with known variance. The signal to noise ratio (SNR) was 10dB, the number of available data was $K = 2000$, and the number of particles was $N = 100$. The value of the parameter α_2 was selected such that $\|\alpha_2 \boldsymbol{d}\| = \min(\|\alpha_1 \boldsymbol{d}\|, 0.01)$, where $\|\alpha\|$ is the length of the vector α. The parameter of the GARCH process is $\boldsymbol{\theta} = [2.00\ 0.50\ 0.20]$. For evaluating the performance of our method we used the normalized root means quare error (NRMSE) in parameter estimation which is obtained using 2000 Monte-Carlo realizations. The simulation results are depicted in Table 8.2. It is apparent that the proposed method which considers the stationarity condition has a better performance than that of competing methods except MLC which is not applicable to noisy data. The main drawback of our method is its high computational load. Some techniques are available to decrease this computational load drastically. For more information on these methods, see [55].

8.4 Conclusions

We have proposed an algorithm based on particle methods for parameter estimation of ARCH and GARCH models in presence of additive noise. The main advantages of our method over the competing parameter estimation methods are dealing with contaminating noise and non-Gaussian distributions of the process noise and corrupting noise. The proposed method also guarantees a stationary model. Our method can be used in other state space parameter estimation problems having linear constraints. It is worth mentioning that the proposed method can be generalized such that it handles nonlinear constraints as well. The main drawback of our method is its high computational load. Some techniques are available to decrease this computational load.

Chapter 9

Research Summary and Future Direction

9.1 Research Summary

In this thesis, we have investigated two different approaches in signal processing. In the first part of this dissertation, we have chosen geometrically originated approach in data processing. We used these methods to introduce a voice activity detection algorithm in presence of transient noise. We have also used these methods to propose an embedding and function extension algorithm for directed graph utilizing Laplace operator. We have also presented a method for extending a bandlimited function defined on a manifold.

In the second part of the thesis, we utilized parametric model-based approaches for processing speech and image data. More specifically, we used the ARCH and GARCH models and their extension in voice activity detection and anomaly detection in SONAR images. We also presented two parameter estimation algorithms for estimating the parameters of these model in presence of additive noise. The main contributions of the thesis chapters are as follows:

In Chapter 2, we proposed a novel voice activity detector based on spectral clustering method. Our main concern was dealing with transient noise, which is very challenging to handle. Almost all conventional methods fail in this situation. Our VAD is a supervised learning algorithm that requires some training data in order to estimate the parameters of the kernel used for computation of a similarity matrix. We used GMM to model the eigenvectors of the similarity matrix. In the testing stage, we used eigenvector extension and proposed a VAD which can be used for online processing of the data with a small delay. Simulation results have demonstrated the high performance of the proposed method, particularly its advantage in

treating transient noises.

In Chapter 3, we proposed a novel method based on Laplace operator for embedding a directed graph. We modeled the observed graph as a sample from a manifold endowed with a vector field, and designed an algorithm that separates and recovers the features of this process: the geometry of the manifold, the data density and the vector field. More specifically, we introduced an anisotropic diffusion type operator and we showed that under certain conditions this operator converges to a Laplace-Beltrami operator on the manifold. Using the fact that eigenfunctions of Laplace-Beltrami operator can be regarded as coordinates on the data set, we derived our embedding procedure. We also proposed a Nystöm type embedding extension and a novel technique for extending an empirical function on a directed graph. Simulation results demonstrated the advantage of the proposed method in function extension on a directed graph.

In Chapter 4, we proposed a novel technique for extending a bandlimited function on a homogenous manifold while the function and manifold are both unknown. The only assumption for retrieving the function is that we have enough data to learn the function (i.e. enough points on the manifold with known function values) and enough data to learn the manifold (i.e. enough samples on the manifold where the value of function is either known or unknown). Diffusion maps were utilized in order to learn the manifold (i.e. approximating the Laplace-Beltrami operator on the manifold). Simulation results show the advantage of the proposed method over the widely-used geometric harmonics. The proposed method is insensitive to the sampling procedure on the manifold. More specifically, the proposed method is not sensitive to the distribution of points sampled on the manifold. This is related to the property that diffusion maps with appropriate normalization recoverd the Riemannian geometry of the data set. An important issue which must be addressed here is the selection of the constant c_0 (see (c) in Chpater 4). This constant is manifold-dependent and must be chosen appropriately so that perfect reconstruction is possible. Finding a way to choose this constant properly requires further research.

In Chapter 5, we presented a new ML estimation procedure for parameter estimation of complex GARCH$(1,1)$ model in presence of the additive noise, and a new procedure for estimating the hidden states of the model in the MMSE sense. The method computes the log-likelihood function together with its gradient vector and Hessian matrix, and uses the Newton method (or gradient projection method under stationarity conditions) to numerically find the maximum of the log-likelihood function. We also presented an RML method for estimating the time varying

parameters. Simulation results show improvement with respect to not considering the noise effect in the estimation of the parameters. In this work we considered two situations. The first one is the general case where the SNR is unknown. In this situation, the proposed method estimates the variance of the noise together with the parameters and the hidden states of the model. The second situation, which is a special case of the first one, assumes that the variance of the additive noise is known and just estimates the parameters of the model. As expected, the second case yields better performance over the first one because of the additional information employed. The issue how to handle colored noise is also discussed in the chapter.

In Chapter 6, we presented a new VAD based on AR(p)-GARCH $(1,1)$ modeling of the speech signal. We also introduced a novel procedure based on the ML estimation method for parameter estimation of the AR(p)$-$GARCH $(1,1)$ model in presence of additive noise. We presented an adaptive version of parameter estimation method, namely, the RML method. In this method, upon receiving a new sample, we updated the likelihood function together with its gradient vector and used the steepest descent method to numerically find the maximum of the log-likelihood function. The proposed VAD is obtained by comparing the likelihood ratio to a threshold. The likelihood ratio of the observations was computed using the estimated variance of the speech signal, which was obtained by the RML algorithm. We used the IMCRA method for estimating the variance of the additive noise, which can be used under nonstationary conditions. This enabled our VAD to follow variations in the variance of the additive noise. Simulation results have demonstrated the high performance of the proposed method and particularly its advantage in nonstationary noise environments.

In Chapter 7, we used the wavelet transform along with a noncausal AR-ARCH model for anomaly detection in side scan sonar images. We introduced a novel background model (i.e. noncausal AR-ARCH model) and used matched subspace detector (MSD) to detect an anomaly in sonar images. We introduced the two dimensional noncausal ARCH model and obtained sufficient stationarity conditions. We also proposed the TSLS estimator for parameter estimation of this model. This estimator has a closed form obtained by solving two sets of linear equations. We have shown that this estimator is asymptotically consistent. The most important advantage of this parameter estimation method is its low computational complexity. We also presented an anomaly detection algorithm based on noncausal AR-ARCH modeling of the background. The detection algorithm utilizes a matched subspace detector [101] for the

background noncausal AR-ARCH model. This model is an extension of the GARCH model used in [91] for anomaly detection. The advantages of the proposed method over that presented in [91] are lower computational load and lower dependency on the orientation of the image. Simulation results have demonstrated the performance of the proposed method.

In Chapter 8, we proposed an algorithm based on particle methods for parameter estimation of ARCH and GARCH models in presence of additive noise. The main advantages of our method over the competing parameter estimation methods are dealing with contaminating noise and non-Gaussian distributions of the process noise and corrupting noise. The proposed method also guarantees a stationary model. Our method can be used in other state space parameter estimation problems having linear constraints. It is worth mentioning that the proposed method can be generalized such that it handles nonlinear constraints as well. The main drawback of our method is its high computational load. Some techniques are available to decrease this computational load.

9.2 Future Research Directions

In this thesis, we introduced two different approaches for data processing. In the first part, we used a geometric approach and developed a voice activity detection algorithm. The same approach can be used in other speech processing algorithms. More specifically, one can utilize these techniques in speaker diarisation which is the process of partitioning an input audio stream into homogeneous segments according to the speaker identity. Speaker diarisation can enhance the readability of an automatic speech transcription by structuring the audio stream into speaker turns and, when used together with speaker recognition systems, by providing the speakers true identity. It is used to answer the question *who spoke when*. Speaker diarisation is a combination of speaker segmentation and speaker clustering. The first aims at finding speaker change points in an audio stream. The second aims at grouping together speech segments on the basis of speaker characteristics.

We presented a novel technique for directed graph embedding. The new results can be applied to various problems in which the similarity between points is not symmetric. Among several examples in which the similarity between points has not to be necessarily symmetric, one can mention document classification based on citation graphs, web information retrieval

based on hyperlink structure, and protein clustering based on pair-wise alignment scores. It is worth investigating the proposed algorithm in such circumstances.

We have provided a novel technique for out-of-sample band-limited function extension for on a homogeneous manifold while the function and manifold are both unknown. Among other applications, this method can be used for supervised speech recognition, speaker recognition and speaker diarisation. An important issue which is worth investigating is the selection of the constant c_0 (see Chapter 4). This constant is manifold-dependent and must be chosen appropriately so that perfect reconstruction is possible. Finding a way to choose this constant properly requires further research.

We also introduced a novel model (i.e. *noncausal* AR-ARCH) for modelling the background in SONAR images. The method proposed for analysing this model, can be used for introducing different *noncausal* model in image processing. Furthermore, this model might be useful in image denoising or deblurring.

Bibliography

[1] [online]. available: http://www.freesound.org.

[2] Abdolahi, M. and Amindavar, H. (2005) GARCH coefficients as feature for speech recognition in persian isolated digit. *Proc. 30th IEEE Int. Conf. Acoust., Speech Signal Process.,(ICASSP)*, pp. 957–960.

[3] Abramson, A. and Cohen, I. (2006) Markov-switching GARCH model and application to speech enhancement in subbands. *Proc. 10th International Workshop on Acoustic Echo and Noise Control, IWAENC-2006, Paris, France, paper no. 7, pp. 1-4.*.

[4] Abramson, A. and Cohen, I. (2006) State smoothing in markov-switching time-frequency GARCH models. *IEEE Signal Process. Lett.*, **13**, 377–380.

[5] Abramson, A. and Cohen, I. (2007) On the stationarity of markov-switching GARCH processes. *Econometric Theory*, **23, no. 3**, 485–500.

[6] Abramson, A. and Cohen, I. (2008) Single-sensor audio source separation using classification and estimation approach and GARCH modeling. *IEEE Trans. Audio, Speech and Language Processing*, **16, no. 8**, 1528–1540.

[7] Abramson, A., Habets, E. A. P., Gannot, S., and Cohen, I. (2008) Dual-microphone speech dereverberation using GARCH modeling. *Proc. 33rd IEEE Internat. Conf. Acoust. Speech Signal Process., ICASSP-2008, Las Vegas, Nevada, pp. 4565-4568.*.

[8] Allen, J. B. and Berkley, D. A. (1979) Image method for efficiently simulating small-room acoustics. *The Journal of the Acoustical Society of America*, **65**, 943.

[9] Amirmazlaghani, M. and Amindavar, H. (2007) Image denoising using two-dimensional garch model. *Systems, Signals and Image Processing, 2007 and 6th EURASIP Confer-*

ence focused on Speech and Image Processing, Multimedia Communications and Services. 14th International Workshop on, pp. 397–400, IEEE.

[10] Amirmazlaghani, M., Amindavar, H., and Moghaddamjoo, A. (2009) Speckle suppression in SAR images using the 2-D garch model. *Image Processing, IEEE Transactions on*, **18**, 250–259.

[11] Ashton, E. A. (1998) Detection of subpixel anomalies in multispectral infrared imagery using an adaptive bayesian classifier. *IEEE Trans. Geoscience and Remote Sensing*, **36**, 506–517.

[12] Bach, F. R. and Jordan, M. I. (2006) Learning spectral clustering, with application to speech separation. *Journal of Machine Learning Research*, **7**, 1963–2001.

[13] Basbug, F., Swaminathan, K., and Nandkumar, S. (2004) Noise reduction and echo cancellation front-end for speech codecs. *IEEE Trans. Speech and Audio Processing*, **11, no.1**, 1–13.

[14] Belkin, M. and Niyogi, P. (2003) Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, **15**, 1373–1396.

[15] Benesty, J., Makino, S., and Chen, J. (2005) *Speech Enhancement*. Springer, Berlin, Germany.

[16] Benyassine, A., Shlomot, E., Su, H., Lamblin, D. M. C., and Petit, J. (1997) `ITU-T` recommendation `G.729x` annex `B`: a silence compression scheme for use with `G.729` optimized for V.70 digital simultaneous voice and data applications. *IEEE Commun. Mag.*, **35**, 64–73.

[17] Bilmes, J. (1998) A gentle tutorial of the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Tech. rep., International Computer Science Institute Berkeley CA.

[18] Bishop, C. (2006) *Pattern recognition and machine learning*. Springer.

[19] Bollerslev, T. (1986) Generalized autoregressive conditional heteroscedasticity. *Journal of Econometrics*, **31**, 307–327.

[20] Bose, A. and Mukherjee, K. (2003) Estimating the ARCH parameters by solving linear equations. *Journal of Time series analysis*, **24**, 127–136.

[21] Chang, J. H. and Kim, N. S. (2003) Voice activity detection based on complex laplacian model. *Electron. Lett.*, **39, no. 7**, 632–634.

[22] Chen, T., Yang, Q., and Tang, X. (2007) Directed graph embedding. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2707–2712.

[23] Cohen, I. (2003) Noise spectrum estimation in adverse environments: improved minima controlled recursive averaging. *IEEE Trans. Speech and Audio Processing*, **11**, 466–475.

[24] Cohen, I. (2005) Speech enhancement using supergaussian speech models and noncausal a priori snr estimation. *Speech Communication*, **47, no. 3**, 336–350.

[25] Cohen, I. (2006) Speech spectral modeling and enhancement based on autoregressive conditional heteroscedasticity models. *signal processing*, **86**, 698–709.

[26] Cohen, I. and Berdugo, B. (2001) Speech enhancement for non-stationary noise environments. *Signal Processing*, **81, no. 11**, 2403–2418.

[27] Coifman, R. and Lafon, S. (2006) Geometic harmonics: A novel tool for multiscale out-of-sample extension of empirical functions. *Appl. Comput. Harmon. anal.*, **21**, 3117.

[28] Coifman, R. and Lafon, S. (2006) Geometic harmonics: A novel tool for multiscale out-of-sample extension of empirical functions. *Appl. Comput. Harmon. anal.*, **21**, 31–52.

[29] Coifman, R., Lafon, S., Lee, A. B., Maggioni, M., Nadler, B., Warner, F., and Zucker, S. W. (2005) Geometric diffusions as a tool for harmonic analysis and structure definition of data: diffusion maps. *Proceedings of the National Academy of Sciences of the United States of America*, **102**, 7426–7431.

[30] Coifman, R., Shkolnisky, Y., Sigworth, F. J., and Singer, A. (2008) Graph laplacian tomography from unknown random projections. *IEEE Trans. Image Processing*, **17**, 1891–1899.

[31] Coifman, R. R. and Hirn, M. J. (2013) Bi-stochastic kernels via asymmetric affinity functions. *Applied and Computational Harmonic Analysis*.

[32] Coifman, R. R. and Lafon, S. (2006) Diffusion maps. *Appl. Comput. Harmon. Anal.*, **21**, 5–30.

[33] Coifman, R. R. and Lafon, S. (2006) Geometric harmonics: A novel tool for multiscale out-of-sample extension of empirical functions. *Appl. Comput. Harmon. anal.*, **21**, 31–52.

[34] Coifman, R. R., Lafon, S., Lee, A. B., Maggioni, M., Warner, F., and Zucker, S. (2005) Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the National Academy of Sciences*, pp. 7426–7431.

[35] Coifman, R. R., Shkolnisky, Y., Sigworth, F. J., and Singer, A. (2008) Graph laplacian tomography from unknown random projections. *IEEE Trans. Image Processing*, **17**, 1891–1899.

[36] Cornu, E., Sheikhzadeh, H., Brennan, R., Abutalebi, H., Tam, E., Iles, P., and Wong, K. (2003) ETSI AMR-2 VAD: evaluation and ultra low-resource implementation. *Proc. Inter. Conf. Multi. Expo, (ICME'03)*, vol. 2, pp. 841–844, IEEE.

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

[38] Diaconis, P., Holmes, S., Shahshahani, M., et al. (2013) Sampling from a manifold. *Advances in Modern Statistical Theory and Applications: A Festschrift in honor of Morris L. Eaton*, pp. 102–125, Institute of Mathematical Statistics.

[39] Dobeck, G. and Hyland, J. (1995) Sea mine detection and classification using side-looking sonars. *Proceedings of the SPIE Annual International Symposium on Aerospace/Defense Sensing, Simulation andControl*, vol. 2496.

[40] Donoho, D. L. and Grimes, C. (2003) Hessian eigenmaps: New locally linear embedding techniques for high-dimensional data. *PNAS*, **100**, 5591–5596.

[41] Donoho, D. L. and Grimes, C. (2005) Image manifolds which are isometric to euclidean space. *J. Math. Imaging Vis.*, **23**, 5–24.

[42] Engle, R. (1982) Autoregressive conditional heteroskedasticity with estimates of the variance of u.k. inflation. *Econometrica*, **50**, 987–1008.

[43] Ephraim, Y. and Malah, D. (1984) Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator. *IEEE Trans. Acoust., Speech, Signal Processing*, **32, no. 6**, 1109–1121.

[44] Ephraim, Y. and Malah, D. (1985) Speech enhancement using a minimum mean-square error log-spectral amplitude estimator. *IEEE Trans. Acoust., Speech, Signal Processing*, **32**, 443175.

[45] Espi, M., Fujimoto, M., Saito, D., Ono, N., and Sagayama, S. (2012) A tandem connectionist model using combination of multi-scale spectro-temporal features for acoustic event detection. *Proc. of ICASSP '12, pp. 4293-4296, March 2012.*.

[46] Fowlkes, C., Belongie, S., Chung, F., and Malik, J. (2004) Spectral grouping using the Nyström method. *IEEE Trans. Patt. Analy. and Mach. Intell.*, **26**, 214 – 225.

[47] Francos, R. P., Permuter, H., and Francos, J. (2003) Gaussian mixture models of texture and colour for image database. *in Proc . ICASSP*, pp. 25–88.

[48] Fukunaga, K. (1990) *Introduction to statistical pattern recognition (2nd ed.)*. Academic Press Professional, Inc.

[49] Garner, N. R., Barrett, P. A., Howard, D. M., and Tyrrell, A. M. (1997) Robust noise detection for speech detection and enhancement. *Electron. Lett.*, **33**, 270–271.

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

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

[52] Gepshtein, S. and Keller, Y. (2013) Image completion by diffusion maps and spectral relaxation. *IEEE Transactions on Image Processing*, **22**, 2983 – 2994.

[53] Giles, C. L., Bollacker, K. D., and Lawrence, S. (1998) Citeseer: An automatic citation indexing system. *Proceedings of the third ACM conference on Digital libraries*, pp. 89–98, ACM.

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

[55] Gray, A. G. (2003) Nonparametric density estimation: toward computational tractability. *SIAM International Conference on Data Mining*, pp. 203–211.

[56] Habets, E. A. P. (2010), Room impulse response (RIR) generator.

[57] Haykin, S. S. (2001) *Kalman Filtering and Neural Networks*. John Wiley & Sons, Inc., New York, NY, USA.

[58] Hein, M. and yves Audibert, J. (2005) Intrinsic dimensionality estimation of submanifolds in R^d. *ICML*, pp. 289–296.

[59] Hoff, P. D., Raftery, A. E., and Handcock, M. S. (2002) Latent space approaches to social network analysis. *Journal of the american Statistical association*, **97**, 1090–1098.

[60] Huang, P. S., Zhuang, X., and Hasegawa-Johnson, M. (2011) Improving acoustic event detection using generalizable visual features and multi-modality modeling. *ICASSP*, pp. 349–352, IEEE.

[61] Ishizuka, K., Nakatani, T., Fujimoto, M., and Miyazaki, N. (2010) Noise robust voice activity detection based on periodic to aperiodic component ratio. *Speech Commun.*, **52**, 41–60.

[62] Jain, A. K. and Jain, J. R. (1978) Partial differential equations and finite difference methods in image processing, part ii: Image restoration. *IEEE Trans. Automat. Conrr*, **23**, 817– 834.

[63] Jansen, A. and Niyogi, P. (2005) A geometric perspective on speech sounds. Tech. rep., Computer Science Dept., Univ. of Chicago.

[64] Kadec, M. I. (1964) The exact value of the paley-wiener constant. *Soviet Math. Doklady,*, **5**, 559–561.

[65] Karray, L. and Martin, A. (2003) Toward improving speech detection robustness for speech recognition in adverse environments. *Speech Communication*, **3**, 261–276.

[66] Kay, S. M. (1993) *Fundamentals of Statistical Signal Processing: Estimation*. Prentice Hall Signal Processing Series.

[67] Kazantsev, I. G., Lemahieu, I., Salov, G. I., and Denys, R. (2002) Statistical detection of defects in radiographic images in nondestructive testing. *Signal Processing*, **82**, 791–801.

[68] Kraut, S., Scharf, L. L., and McWhorter, L. T. (2001) Adaptive subspace detectors. *IEEE Trans. Signal Processing*, **49**, 1–16.

[69] Kreyszig, E. (1989) *Introductory Functional Analysis with Applications*. Wiley; (first edition).

[70] Kushnir, D., Haddad, A., and Coifman, R. (in press, 2011) Anisotropic diffusion on submanifolds with application to earth structureclassification. *Appl. Comput. Harmon. anal.*.

[71] Lafon, S., Keller, Y., and Coifman, R. R. (2006) Data fusion and multicue data matching by diffusion maps. *IEEE Trans. Patt. Analy. and Mach. Intell.*, **28**, 1784–1797.

[72] Lafon, S. and Lee, A. B. (2006) Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *IEEE Trans. Patt. Analy. and Mach. Intell.*, **28**, 1393–1403.

[73] Laine, A. F., Schuler, S., Fan, J., and Huda., W. (1994) Mammographic feature enhancement by multiscale analysis. *IEEE Transactions on Medical Imaging,*, **13**, 725–740,.

[74] Li, W., Ling, S., and McAleer, M. (2002) Recent theoretical results for time series models with GARCH errors. *Journal of Economic Surveys*, **16**, 245–269.

[75] Ljung, L. and SAoderstrAom, T. (1983) *Theory and Practice of Recursive Identification*. MIT Press, Cambridge.

[76] Luenberger, D. G. (1989) *Linear and nonlinear programming*. Addison- Wesley Publishing Company, Inc.

[77] Luxburg, U. V. (2007) A tutorial on spectral clustering. *Statistics And Computing*, **17**, 395–416.

[78] Lyubarskii, Y. and Madych, W. R. (1994) The recovery of irregularly sampled band limited functions via tempered splines. *Journal of Functional Analysis*, **125**, 201–222.

[79] Manolakis, D. and Shaw, G. (2002) etection algorithms for hyperspectral imaging applications. *IEEE Signal Process. Mag.*, **19**, 29–43.

[80] McClellan, J. H. (1982) Multidimensional spectral estimation. *Proc. IEEE,*, **70**, 1029–1039.

[81] Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2012) *Foundations of machine learning*. The MIT Press.

[82] Mousazadeh, S. and Cohen, I. (2010) Anomaly detection in sonar images based on wavelet domain noncausal AR-ARCH random field modeling. *Proc. 26th IEEE Convention of Electrical and Electronics Engineers in Israel, IEEEI-2010, Eilat, Israel, 17-20 November 2010.*.

[83] Mousazadeh, S. and Cohen, I. (2010) Simultaneous parameter estimation and state smoothing of complex GARCH process in the presence of additive noise. *signal processing*, **90-11**, 2947–2953.

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

[85] Mousazadeh, S. and Cohen, I. (2013) Voice activity detection in presence of transient noise using spectral clustering. *IEEE Transactions on Audio, Speech & Language Processing*, **21**, 1261–1271.

[86] Mousazadeh, S. and Karimi, M. (2007) ARCH parameter estimation via constrained two stage least squares method. *Proc .IEEE International Symposium on Signal Processing and its Application, (ISSPA)*, pp. 1–4.

[87] Mousazadeh, S. and Karimi, M. (2009) Estimating multivariate ARCH parameters by two-stage least-squares method. *Signal Processing*, **89**, 921–932.

[88] Nadler, B., Lafon, S., Coifman, R. R., and Kevrekidis, I. G. (2006) Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. *Applied and Computational Harmonic Analysis*, **21**, 113–127.

[89] Nadler, B., Srebro, N., and Zhou, X. (2009) Statistical analysis of semi-supervised learning: The limit of infinite unlabelled data. *Advances in Neural Information Processing Systems*, pp. 1330–1338.

[90] Ng, A. Y., Jordan, M. I., and Weiss, Y. (2001) On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing System*, pp. 849–856, MIT Press.

[91] Noiboar, A. and Cohen, I. (2007) Anomaly detection based on wavelet domain GARCH random field modeling. *IEEE Trans. Geoscience and Remote Sensing*, **45**, 1361–1373.

[92] Pentney, W. and Meila, M. (2005) Spectral clustering of biological sequence data. Veloso, M. M. and Kambhampati, S. (eds.), *AAAI*, pp. 845–850, AAAI Press / The MIT Press.

[93] Perrault-Joncas, D. and Meilă, M. (2011) Directed graph embedding: an algorithm based on continuous limits of laplacian-type operators. *Advances in Neural Information Processing Systems*, **25**.

[94] Pesenson, I. (2000) A Sampling Theorem on Homogeneous Manifolds. *Transactions of the American Mathematical Society*, **352**, 4257–4269.

[95] Poyiadjis, G., Doucet, A., and Singh, S. S. (2005) Particle methods for optimal filter derivative: Application to parameter estimation. *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP)*, pp. v/925– v/928.

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

[97] Ramrez, J., Segura, J., Bentez, C., de la Torre, A., and Rubio, A. (2003) A new adaptive longterm spectral estimation voice activity detector. *Proc. EUROSPEECH 2003, Geneva, Switzerland, pp. 3041-3044*.

[98] Reynolds, D. A. and Rose, R. C. (1995) Robust text-independent speaker identification using gaussian mixture speaker models. *IEEE Trans. Speech and Audio Processing*, **3**, 72 – 83.

[99] Roweis, S. T. and Saul, L. K. (2000) Nonlinear dimensionality reduction by locally linear embedding. *Science*, **290, no. 5500**, 2323–2326.

[100] Sangwan, A., Chiranth, M., Jamadagni, H., Sah, R., and Gaurav, R. P. V. (2002) VAD techniques for real-time speech transmission on the Internet,. *IEEE International Conference on High-Speed Networks and Multimedia Communications, pp. 46-50.*.

[101] Scharf, L. L. and Friedlander, B. (1994) Matched subspace detectors. *IEEE Trans. Signal Processing*, **42**, 2146–2157.

[102] Schoenberg, I. (1973) *Cardinal Spline Interpolation*. CBMS, 12, SIAM, Philadelphia.

[103] Scholkopf, B., Smola, A., and Muller, K. (1996) Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, **10**, 1299–1319.

[104] Schweizer, S. M. and Moura, J. M. F. (2002) Hyperspectral imagery: Clutter adaptation in anomaly detection. *IEEE Trans. Information Theory*, **46**, 1855–1871.

[105] Shadrin, A. Y. (2001) The L^{∞}-norm of thel 2-spline projector is bounded independently of the knot sequence: A proof of de boor's conjecture. *Acta Mathematica*, **187**, 59–137.

[106] Shannon, C. E. (1949) Communication in the Presence of Noise. *Proceedings of the IRE*, **37**, 10–21.

[107] Shi, J. and Malik, J. (2000) Normalized cuts and image segmentation. *IEEE Trans. Patt. Analy. and Mach. Intell.*, **22**, 888–905.

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

[109] Singer, A. (2006) Spectral independent component analysis. *Applied and Computational Harmonic Analysis*, **21**, 135–144.

[110] Singer, A. and Coifmain, R. (2008) Non-linear independent component analysis with diffusion maps. *Appl. Comput. Harmon. anal.*, **25, no. 2**, 226–239.

[111] Sohn, J., Kim, N. S., and Sung, W. (1999) A statistical model-based voice activity detection. *IEEE Signal Process. Lett.*, **16**, 1–3.

[112] Solvanga, H. K., Ishizukac, K., and Fujimotoc, M. (2008) Voice activity detection based on adjustable linear prediction and GARCH models. *Speech Communication*, **50**, 476–486.

[113] Spence, C., Parra, L., and Sajda, P. (2001) Detection, synthesis and compression in mammographic image analysis with a hierarchical image probability model. *Proceedings of the IEEE Workshop on Mathematical Methods in Biomedical Image Analysis. IEEE Computer Society*.

[114] Stein, D. W. J., Beaven, S. G., Hoff, L. E., Winter, E. M., Schaum, A. P., and Stocker, A. D. (2002) Anomaly detection from hyperspectral imagery. *IEEE Signal Process. Mag.*, **19**, 58–69.

[115] Stoica, P. and Moses, R. L. (1997) *Introduction to Spectral Analysis*. Prentice Hall.

[116] Strickland, R. N. and Hahn, H. I. (1997.) Wavelet transform methods for object detection and recovery. *IEEE Trans. Image Process.*, **6**, 724175.

[117] Tadic, V. and Doucet, A. (2005) Exponential forgetting and geometric ergodicity for optimal filtering in general state-space models. *Stochastic Processes and Their Applications*, **115**, 1408–1436.

[118] Tahmasbi, R. and Rezaei, S. (2007) A soft voice activity detection using GARCH filter and variance gamma distribution. *IEEE Trans. Audio, Speech and Language Processing*, **5**, 1129–1134.

[119] Talmon, R., Cohen, I., and Gannot, S. (2011) Supervised source localization using diffusion kernels. *Proc. 2011 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA-2011, New Paltz, NY, 16-19*.

[120] Talmon, R., Cohen, I., and Gannot, S. (2011) Transient noise reduction using nonlocal diffusion filters. *IEEE Trans. Audio, Speech and Language Processing*, **19**, 1584–1599.

[121] Talmon, R., Kushnir, D., Coifman, R. R., Cohen, I., and Gannot, S. (2012) Parametrization of linear systems using diffusion kernels. *IEEE Transactions on Signal Processing*, **60**, 1159–1173.

[122] Tekalp, A. M., Kaufman, H., and Woods, J. W. (1986) Identification of image and blur parameters for the restoration of noncausal blurs. *IEEE Trans. Acoust., Speech, Signal Processing*, **34**, 963– 972.

[123] Tenenbaum, J. B., de Silva, V., and Langford, J. C. (2000) A global geometric framework for nonlinear dimensionality reduction. *Sience*, **290**, 2319–2323.

[124] Therrien, C. W., Quatieri, T. F., and Dudgeon, D. E. (1986) Statistical model-based algorithms for image analysis. *Proc. IEEE*, **74**, 532–607.

[125] Ting, D., Huang, L., and Jordan, M. (2011) An analysis of the convergence of graph laplacians. *arXiv preprint arXiv:1101.5435*.

[126] Tuan, P. D. (1988) Maximum likelihood estimation of the autoregressive model by relaxation on the reflection coefficients. *IEEE Trans. Acoust., Speech, Signal Processing*, **36**, 1363–1367.

[127] Volfin, I. and Cohen, I. (2012) Dominant speaker identification for multipoint videoconferencing. *Computer Speech and Language*.

[128] Walnut, D. F. (2004) *An Introduction to Wavelet Analysis*. Birkhäuser Boston.

[129] Weiss, A. A. (1989) Asymptotic theory of ARCH models estimation and testing. *Econometric Theory*, **2**, 107–131.

[130] Willsky, A. S. (2002) Multiresolution markov models for signal and image processing. *Proceedings of the IEEE*, **90**, 1396–1458.

[131] Xia, X. G., Boncelet, C. G., and Arce, G. R. (1997) A multiresolution watermark for digital images. *Proceedings of the International Conference on Image Processing (ICIP '97)*, pp. 497–511, IEEE Computer Society, Washington DC, USA.

[132] Yu, C., Ooi, B., Tan, K., and Jagadish, H. (2001) Indexing the distance: An efficient method to knn processing. *Proceedings of the international conference on very large data bases*, pp. 421–430.

[133] Zelnik-Manor, L. and Perona, P. (2005) Self-tuning spectral clustering. *Advances in Neural Information Processing Systems*, **17**, 1601–1608.

[134] Zhang, Z. and Zha, H. (2002) Principal manifolds and nonlinear dimension reduction via local tangent space alignment. *arXiv preprint cs/0212008*.

[135] Zhao, P. Y. and Yu, D. R. (1993) An unbiased and computationally efficient LS estimation method for identifying parameters of 2-d noncausal SAR models. *IEEE Trans. Signal Processing*, **41**, 849–857.

[136] Zhou, X. and Belkin, M. (2011) Semi-supervised learning by higher order regularization. *International Conference on Artificial Intelligence and Statistics*, pp. 892–900.

[137] Zhou, X., Zhuang, X., Liu, M., Tang, H., Hasegawa-Johnson, M., and Huang, T. (2008) *Multimodal Technologies for Perception of Humans*. Springer-Verlag, Berlin, Heidelberg.

[138] Zhu, X., Ghahramani, Z., and Lafferty, J. (2003) Semi-supervised learning using gaussian fields and harmonic functions. *Int. Conf. Mach. Learn.*, vol. 3, pp. 912–919.

[139] Zontak, M. and Cohen, I. (2009) Defect detection in patterned wafers using multichannel scanning electron microscope. *Signal Processing*, **89, no. 8**, 1511–1520.

[140] Zontak, M. and Cohen, I. (2010) Defect detection in patterned wafers using anisotropic kernels. *Machine Vision and Applications*, **21, no. 2**, 129–141.

עיבוד אותות דיבור וקול מבוסס על אנליזה הרמונית ומודלים סטטיסטים

סאמאן מוסזאדה

עיבוד אותות דיבור וקול מבוסס על אנליזה הרמונית ומודלים סטטיסטים

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר
דוקטור לפילוסופיה

סאמאן מוסזאדה

הוגש לסנט הטכניון - מכון טכנולוגי לישראל

אדר א, תשע״ו ,חיפה , פברואר 2016

המחקר נעשה בהנחיית פרופ׳ ישראל כהן
מהפקולטה להנדסת חשמל בטכניון

תודות

ברצוני להביע את תודתי והערכתי העמוקה לפרופ׳ ישראל כהן
על הדרכתו ופיקוחו המסור. אני מעריך מאד את תמיכתו המקצועית, עידודו לשלמות ותרומתו להתפתחותי האישית, שנוגעים הרבה מעבר להיקף עבודה זו.

כמו כן, ברצוני להודות לפרופ׳ דוד מלאך
על הערותיו מאירות העיניים ועצותיו המועילות.
בנוסף, אני רוצה להודות לנמרוד פלג ואבי רוזן
על התמיכה רבת הערך שלהם.

אודה גם לחבריי ועמיתיי בטכניון. תודה מיוחדת לעידן רם, רונן טלמון, תומר פלג ודוד דב
על תמיכתם וידידותם, על הערותיהם המועילות, ועל כך שבזכותם זמן שהותי בטכניון היה מהנה ובלתי נשכח.

אני רוצה להודות לאלהה.
התמיכה, העידוד, הסבלנות השקטה ואהבתה הבלתי מתפשרת היו ללא ספק הסלע שעליו חלק מהחיים שלי נבנו. הסובלנות שלה היא עדות בפני עצמה למסירות הבלתי מתפשרת שלה ולאהבתה.

והכי חשוב ברצוני להביע את תודתי העמוקה למשפחה היקרה שלי
הורי בנימין ואסתר והאחיות שלי דורנה וגולי, על אהבתם, העידוד והתמיכה שלהם.

אחרונה חביבה, תודות לבתי האהובה, נטע. חיוכיה הם הדבר הכי יפה שראיתי אי פעם.

אני מודה לטכניון ולקרן הלאומית למדע (מענק מס׳ 1130/11), לארגון ע״ש אירווין וג׳ואן ג׳ייקובס, לארגון ע״ש גוטווירט׳ ולארגון ע״ש אולנדורף.
על התמיכה הכספית הנדיבה בהשתלמותי.

תקציר

התחום עיבוד אותות דיבור זכה למאמץ מחקרי משמעותי בעשורים האחרונים. למרות זאת, היבטים רבים עודם פתוחים ודורשים מחקר נוסף. עבודה זו עוסקת בתאוריה וביישומים של שיטות גיאומטריות ומבוססות על מודלים בתחום העיבוד אותות.

עבודה זו מחולקת לשני חלקים. בחלקה הראשון, אנו מתרכזים בשיטות גיאומטריות בעיבוד אותות דיבור. ראשית, נטפל בבעיית גילוי פעילות קול (Voice Activity Detection (VAD. מטרת ה VAD - הינה לסווג בין מסגרות דיבור לבין מסגרות בהן אין דיבור (מסגרות אלו כוללות שקט, רעש וכו'). אלגוריתם ה VAD -מהווה אבן בניין מהותית במגוון יישומים בעיבוד אותות דיבור. ביניהם, זיהוי דיבור רובסטי, שידור רציף ,שידור נאום בזמן אמת באינטרנט או הפחתת רעש, ביטול הדהוד בהקשר של טלפוניה ועוד. בעיקר, נתרכז בגילוי פעילות קול בנוכחותו של רעש טרנזיינטי ע"י שיטות גיאומטריות. נפתח אלגוריתם למידה חדשני לביצוע VAD המבוסס על שיטת ה Spectral - Clustering. אלגוריתם זה מפריד את מסגרות הדיבור לשתי קבוצות שונות, כפי שהוזכר לעיל (מסגרות דיבור ומסגרות שאינן כוללות דיבור). האלגוריתם המוצע הינו אלגוריתם הדורש למידה (Supervised Learning Algorithm). בשל כך, מתבצע שימוש בנתונים המופקים משלב האימון לטובת התאמת פרמטרים לשלב הבוחן. בפרט, הפרמטרים המתקבלים בשלב האימון הם הפרמטרים של הגרעין המשמש לבניית מטריצת הדמיון הנדרשת בשיטות ה-Spectral Clustering. הווקטורים העצמיים של הלפלסיאן המנורמל של מטריצת הדמיון ומודל תערובת גאוסיאנים (Gaussian Mixture Model -GMM) מנוצלים בכדי לחשב את יחס הסבירות הנדרש לגילוי פעילות הדיבור. תוצאות הסימולציה מדגימות את יתרונה של השיטה המוצעת בהשוואה לאלגוריתמים קונבנציונליים המבוססים על מודלים סטטיסטיים.

הנושא הבא בתזה הינו שיטות Graph Embedding על גרפים כיווניים. המחקר העוסק בשיטות המבוססות גיאומטריות ולמידה חישובית הוביל לפיתוח אלגוריתמים יעילים עבור יישומים רבים. ביניהם, הפחתת ממדים, Graph Embedding, סיווג, ועוד. מרבית שיטות אלו הינן טכניקות מבוססת על Graph embedding. היתרון בשיטות אלו הינו הפשטות והיכולת לבטא את היחסים המורכבים בין נקודות שונות במרחב המקורי. הרעיון המרכזי באלגוריתמים אלו הינו בנית גרף משוקלל עבור נתונים באופן שבו כל קודקוד של הגרף מייצג נתון אחד והקשת המחברת בין שני קודקודים, מייצגת את הדמיון בין שני נתונים. בהקשר של רשתות (לדוגמא ,רשתות חברתיות), נתונים באופן טבעי מובילים את עצמם למידול גרפי. כמעט בכל השיטות המוכרות, הדמיון בין נתונים מחושב על ידי גרעין חיובי וסימטרי. בחלק מהיישומים המעשיים, הדמיון בין הנקודות אינו בהכרח סימטרי כפי שמניחים לרוב. בתזה זו, אנו מציעים שיטה חדשה למציאת Graph Embedding לגרפים מכוונים.

אנו מניחים כי נתונים נדגמים על יריעה (Manifold) והדמיון בין הנתונים ניתן על ידי גרעין א-סימטרי (Asymmetric Kernel). אנו מספקים אלגוריתם מבוסס Graph Embedding המבוצע על ידי הפעלת אופרטור מסוג Laplacian על היריעה. בנוסף אנו מציגים שיטה להרחבת פונקציות (Function Extension) עבור פונקציות עצמיות מסוג Nystrom המשמשת גם לטובת Embedding עבור נתונים חדשים ולהרחבת פונקציה אמפירית על נתונים חדשים שהתקבלו. לטובת הרחבת הפונקציות העצמיות לנקודות חדשות, אנו מניחים בעבודה זו כי נתונים רק המרחקים בין הנקודות החדשות לנקודות האימון בלבד. תוצאות הסימולציה מדגימות את ביצועי השיטה המוצעת.

למידה מפוקחת (Supervised Learning) הינה משימת למידה חישובית של הסקת פונקציה מנתוני אימון. סוג הנתונים והמאפיינים של הפונקציה תלויים ביישום. אחת מהבעיות הבסיסיות של למידה מפוקחת הינה דגימה ושחזור של פונקציות חסומות סרט (Band-Limited). בתזה זו, אנו מטפלים בבעיה של הרחבת פונקציות חסומות סרט המוגדרות על יריעה (כלומר תחום ההגדרה של הפונקציה הינו יריעה הומוגנית במרחב האוקלידי). אנו מציגים פתרון לבעיה זו במקרה הכללי בו היריעה אינה ידועה. נניח כי לרשותנו מספיק נתונים אשר יאפשרו ללמוד את הפונקציה ואת היריעה (כלומר ללמוד את התחום של הפונקציה). לשיטה המוצעת קיים פתרון סגור אשר מורכב מכפל והיפוך מטריצות. השיטה המוצעת מתכנסת לפתרון האופטימלי, כל עוד ערכי הפונקציה ידועים על סט דגימה מתאים. תוצאות הסימולציה מראות את יתרונה של השיטה המוצעת על שיטת Geometric Harmonics אשר מהווה שיטה מקובלת להרחבת פונקציות.

בחלקה השני של התזה, אנו מתרכזים באלגוריתמים המבוססים על מודלים. לשם כך, אנו בוחרים לעסוק במודל ARCH (Autoregressive Conditional Heteroskedasticity) והכללתו, מודל GARCH.מודל זה הינו כלי רב עוצמה לניתוח וחיזוי של התנודתיות בשווקים פיננסיים. מודל ARCH הוא מודל סטטיסטי שבמפורש ממדל שונות מותנת משתנה בזמן בשימוש בערכים מוחלטים בריבוע, תוך התחשבות בתנודתיות לא אחידה (volatility clustering) ופילוג עם גבנוניות עודפת (excess kurtosis). ראשית, אנו מציגים משערך נראות מקסימלית (Maximum Likelihood) חדש להערכת פרמטרים של מודלי GARCH בנוכחות של רעש אדיטיבי, ומשערך חדש לאמידת המצבים הנסתרים של המודל במובן (Minimum Mean Square Error) MMSE. במסגרת שיטה זו מחשבים את פונקצית לוג-הסבירות ויחד את הוקטור השיפוע שלה ומטריצת הסיאן (Hessian). נעזר בשיטת ניוטון (או שיטת הקרנת שיפוע בתנאי סטציונריות) כדי למצוא את המקסימום של פונקצית לוג-הסבירות באופן חישובי. בנוסף, אנו מציגים שיטת (Recursive Maximum Likelihood) RML לשיערוך פרמטרים משתנים בזמן. תוצאות הסימולציה מראות את השיפור בביצועים המתקבל ע״י שימוש בשיטה זו על פני שיטות אשר אינן לוקחות בחשבון את השפעת הרעש על שערון הפרמטרים. ניתן להבחין בכך בשני המקרים הבאים. המקרה הראשון הוא המקרה הכללי שבו יחס האות לרעש אינו ידוע. במצב זה, השיטה המוצעת משערכת את השונות של הרעש יחד עם הפרמטרים והמצבים

הנסתרים של המודל. במקרה השני, (המהווה מקרה מיוחד של הראשון) מניחים כי השונות של הרעש האדיטיבי הינה ידועה ונותר לשערך את הפרמטרים של המודל. כצפוי, במקרה השני מתקבלים ביצועים טובים יותר מהמקרה הראשון וזאת, בזכות המידע הנוסף שקיים לרשותנו. בנוסף, תחת פרק זה נעסוק גם בהתמודדות עם רעש לא לבן.

כמו כן, בתזה זו אנו מציגים אלגוריתם VAD חדשני המבוסס על מידול AR(p)-GARCH(1,1) של אות הדיבור. בהקשר זה, אנו מציגים תהליך המבוסס על שיטת אומדן ML להערכת הפרמטר של המודל AR(p)-GARCH(1,1) בנוכחות של רעש אדיטיבי. אנו מציגים גרסה מותאמת של שיטת אומדן פרמטר, כלומר ,שיטת RML. בשיטה זו, עם קבלת מדגם חדש, אנו מעדכנים את פונקצית הסבירות יחד עם וקטור השיפוע שלה. לטובת כך, השתמשנו בשיטת ה Gradient Decent-כדי למצוא באופן נומרי את מקסימום פונקצית הסבירות הלוגריתמית. אלגוריתם ה VAD-המוצע מתקבל על ידי השוואת יחס הסבירות לסף. יחס הסבירות לתצפיות מתקבל באמצעות שונויות משוערות של אות הדיבור, אשר התקבלו על ידי אלגוריתם RML. אנו משתמשים בשיטת IMCRA להערכה השונות של הרעש האדטיבי, אשר ניתן להשתמש בה תחת תנאים של אי-סטציונריות. בשל כך, מתקבל VAD העוקב אחרי השתנותו של שונות הרעש האדטיבי. תוצאות הסימולציה מראות ביצועים גבוהים של השיטה המוצעת ובפרט, את יתרונה בסביבות רעש לא-סטציונרי.

כמו כן בתזה זו אנו מציגים מודל סטטיסטי חדש, מודל AR-ARCH אי-סיבתי. אנו מציגים שיטה חדשנית המבוססת על שיטת (Two Stage Least Square) TSLS להערכת הפרמטרים של המודל. בנוסף, אנו מציגים אלגוריתם לזיהוי אנומליה המבוסס על המודלים AR-ARCH אי-סיבתי של הרקע. אלגוריתם הזיהוי הוא Matched Subspace Detector ומודל הרקע הוא מודל AR-ARCH אי-סיבתי. מודל זה הינו הרחבת המודל AR-GARCH לזיהוי שינוי בהתנהגות. יתרונו של המודל המוצע על מודל GARCH סיבתי הוא הפחתת העומס החישובי ורובסטיות לאורינטציה של התמונה. תוצאות הסימולציה מראות שיפור בביצועים כתוצאה משימוש בשיטה זו.

לבסוף, אנו מציעים שיטה חדשה המבוססת על מסנני חלקיקים (Particle Filters) לטובת שיערוך סבירות מרבי (ML) עבור הפרמטרים של מודל ARCH ומודל GARCH. השיטה המוצעת מבוססת על שיטה Gradient Decent ושיטת Active Set המיועדות למקסם את פונקצית הסבירות של הפרמטרים תחת אילוצי סטציונריות. השיפוע של פונקצית הסבירות עבור נתונים בהינתן הפרמטרים של המודל, (אשר נדרשים לצורך אלגוריתם לאופטימיזציה מבוסס שיפוע), נאמד באמצעות שיטות חלקיקים. תוצאות הסימולציה מראות את היתרון של השיטה המוצעת על טכניקות מתחרות.