BLIND DECONVOLUTION TECHNIQUES IN IDENTIFYING FMRI BASED BRAIN ACTIVATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

HALİME İCLAL AKYOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2011

Approval of the thesis:

BLIND DECONVOLUTION TECHNIQUES IN IDENTIFYING FMRI BASED BRAIN ACTIVATION

submitted by **HALİME İCLAL AKYOL** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. İsmet Erkmen _____
Head of Department, **Electrical and Electronics Engineering**

Prof. Dr. Aydan Erkmen _____
Supervisor, **Electrical and Electronics Engineering Dept., METU**

Assist. Prof. Dr. Didem Gökçay _____
Co-Supervisor, **Informatics Institute**, **METU**

Examining Committee Members:

Prof. Dr. Mustafa Kuzuoğlu _____
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Aydan Erkmen _____
Electrical and Electronics Engineering Dept., METU

Assist. Prof. Dr. Didem Gökçay _____
Informatics Institute, METU

Assist. Prof. Dr. Yeşim Serinağaoğlu Doğrusöz _____
Electrical and Electronics Engineering Dept., METU

Assist. Prof. Dr. Mustafa Doğan _____
Control Engineering Dept., Doğuş University

Date: 15.09.2011

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work

Name, Last name: Halime İclal Akyol

Signature:

ABSTRACT

BLIND DECONVOLUTION TECHNIQUES IN IDENTIFYING FMRI BASED BRAIN ACTIVATION

Akyol, Halime İclal
M.S., Department of Electrical and Electronics Engineering
Supervisor : Prof. Dr. Aydan Erkmen
Co-Supervisor : Assist. Prof. Dr. Didem Gökçay

September 2011, 268 pages

In this thesis, we conduct functional Magnetic Resonance Imaging (fMRI) data analysis with the aim of grouping the brain voxels depending on their responsiveness to a neural task. We mathematically treat the fMRI signals as the convolution of the neural stimulus with the hemodynamic response function (HRF). We first estimate a time series including HRFs for each of the observed fMRI signals from a given set and we cluster them in order to identify the groups of brain voxels. The HRF estimation problem is studied within the Bayesian framework through a blind deconvolution algorithm using MAP approach under completely unsupervised and model-free settings, i.e, stimulus is assumed to be unknown and also no particular shape is assumed for the HRF. Only

using a given fMRI signal together with a weak Gaussian prior distribution imposed on HRF favoring ‘smoothness’, our method successfully estimates all the components of our framework: the HRF, the stimulus and the noise process. Then, we propose to use a modified version of Hausdorff distance to detect similarities within the space of HRFs, spectrally transform the data using Laplacian Eigenmaps and finally cluster them through EM clustering. According to our simulations, our method proves to be robust to lag, sampling jitter, quadratic drift and AWGN (Additive White Gaussian Noise). In particular, we obtained 100% sensitivity and specificity in terms of detecting active and passive voxels in our real data experiments. To conclude with, we propose a new framework for a mathematical treatment for voxel-based fMRI data analysis and our findings show that even when the HRF is unpredictable due to variability in cognitive processes, one can still obtain very high quality activation detection through the method proposed in this thesis.

Keywords: Functional Magnetic Resonance Imaging, Hemodynamic Response Function, Blind Deconvolution, Hausdorff Distance

ÖZ

FMRG TABANLI BEYİN AKTİVASYONLARININ SAPTANMASINDA GÖZÜ KAPALI TERS EVRİŞİM TEKNİKLERİ

Akyol, Halime İclal
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Aydan Erkmen
Ortak Tez Yöneticisi : Yrd. Doç. Dr. Didem Gökçay
Eylül 2011, 268 sayfa

Bu tez çalışmasında; benzer beyin voksellerini, belirli uyaranlara verilen tepkilere bağlı olarak gruplandırmak amacıyla, fonksiyonel Manyetik Rezonans Görüntüleme (fMRG) veri incelemesi yapılmıştır. fMRG sinyalleri, sinirsel uyarı ile hemodinamik yanıt fonksiyonunun (HYF) evrişimi olarak düşünülmüş ve hemodinamik yanıt fonksiyonlarını içeren zaman dizilerinin bu sinyaller kullanılarak kestirimi yoluyla, beyin vokselleri gruplandırılmıştır. HYF kestirimi problemi; MAP kestirim yöntemi kullanan bir gözü kapalı ters evrişim algoritması yoluyla Bayesçi bir çerçeve dahilinde çalışılmıştır. Bu yaklaşım; voksellerde oluşan sinirsel uyarılarının bilinmemesi ve HYF üzerine herhangi bir model oturtulmaması dolayısıyla gözetimsiz ve modellerden

bağımsızdır. Bu bağlamda, HYF dağılımının, küçük türevli fonksiyonlara meyil eden bir öncül Gauss dağılımı olduğunu kabul eden ve sadece fMRG sinyallerini girdi olarak kullanan başarılı bir HYF kestirim metodu geliştirilmiştir. Bu metot sadece HYF değil aynı zamanda söz konusu evrişimin diğer bileşenlerini de (sinirsel uyarılar ve gürültü) başarıyla kestirebilmektedir. Bunu takiben, kestirilmiş HYF'lerin arasında düzeltilmiş Hausdorff uzaklığı kullanılarak kurulan benzerlikler sayesinde veriler spektral olarak dönüştürülür ve bunlar EM gruplama yöntemi ile gruplandırılır. Yapılan simülasyonlara göre, geliştirilen metodun zamandaki kaymalara, örnekleme seğirmelerine, bağımsız Gauss gürültüye ve genlik sürüklenmelerine karşı gürbüz olduğu gösterilmiştir. Özellikle, aktif ve aktif olmayan beyin voxellerinin birbirlerinden ayrıştırılması noktasında, yapılan gerçek veri deneylerinde % 100 duyarlılık ve özgüllük tespit edilmiştir. Sonuç olarak bu tez çalışmasında matematiksel bir tabana oturan fMRG veri incelemesi için yeni bir Bayesci çerçeve geliştirilmiş, önerilmiştir. Deney ve simülasyon bulgularına göre, bilişsel süreçlerdeki çeşitlilikler yüzünden HYF tahmin edilemese bile, önerilen metot sayesinde halen yüksek kalitede aktivasyon tespiti mümkündür.

Anahtar kelimeler: Fonksiyonel manyetik rezonans görüntüleme, hemodinamik yanıt fonksiyonu, gözü kapalı ters evrişim, Hausdorff uzaklığı

ACKNOWLEDGEMENTS

I would like to give special thanks to my thesis supervisor, Prof. Dr. Aydan Erkmen, and co-supervisor, Assist. Prof. Dr. Didem Gökçay; for their professional support, guidance and encouragement which were invaluable for me during this thesis' preparation.

I would also like to thank Mr.Ulaş Çiftçioğlu, Mr. Mete Balcı and Mr. Serdar Baltacı for their assistance.

TABLE OF CONTENTS

ABSTRACT ..... **III**

ÖZ ..... **V**

ACKNOWLEDGEMENTS ..... **VII**

TABLE OF CONTENTS ..... **VIII**

LIST OF TABLES ..... **X**

LIST OF FIGURES ..... **XI**

1 INTRODUCTION **1**

1.1 THESIS OBJECTIVE 7

1.2 METHODOLOGY 9

1.3 CONTRIBUTIONS 15

1.4 OUTLINE OF THE THESIS 17

2 LITERATURE SURVEY AND MATHEMATICAL BACKGROUND **19**

2.1 APPROACHES TO FMRI ANALYSIS 22

2.1.1 Hypothesis-Driven Methods 22

2.1.2 Data-Driven Methods 27

2.1.3 Blind Deconvolution Applications 33

2.2 APPROACHES TO CLUSTERING OF FMRI 45

2.2.1 Hierarchical Clustering 45
2.2.2 K-means Clustering....47
2.2.3 Expectation Maximization (EM)....51
2.2.4 Spectral Clustering....55

3 METHODOLOGY....69

3.1 MAXIMUM A POSTERIORI (MAP) BLIND DECONVOLUTION OF FMRI 69
3.2 PREPROCESSING 101
3.3 HAUSDORFF DISTANCE OF FMRI 105
3.4 SPECTRAL CLUSTERING AFTER MAP BLIND DECONVOLUTION 146

4 EXPERIMENTS AND RESULTS....151

4.1 HEMODYNAMIC RESPONSE FUNCTION EXTRACTION....156
HEMODYNAMICAL TIME SERIES....175
4.1.1 Block Design Hemodynamical Time Series 176
4.1.2 Categorical Block Design Hemodynamical Time Series....183
4.2 CLUSTERING RESULTS 199
4.2.1 Clustering Results for Simulated Data....199
4.2.2 Clustering Results for Real Data....224

5 SENSITIVITY AND PERFORMANCE ANALYSIS....230

6 CONCLUSIONS 243

REFERENCES....247

APPENDICES 256

LIST OF TABLES

Table 3.1 The mean and standard deviations of HD under different noise conditions..121
Table 3.2 The mean and standard deviations of standard HD under different noise conditions 127
Table 3.3 The mean and standard deviations of modified HD with α=0.99 under different noise conditions 127
Table 3.4 The mean and standard deviations of modified HD with α=0.97 under different noise conditions 128
Table 3.5 The mean and standard deviations of modified HD with α=0.95 under different noise conditions 128
Table 4.1 Clustering results using and BOLD response and the raw fMRI.....214
Table 4.2 Comparison of different distance and similarity measures 215
Table 4.3 Clustering results under different noise conditions.....216
Table 4.4 Clustering results under combined noise and lag-drift conditions.....223
Table 5.1 Sensitivity and specificity of the algorithm under different choices for p.....232
Table 5.2 Sensitivity and specificity under different κ values 234
Table 5.3 Sensitivity and specificity under different α values 236
Table 5.4 Sensitivity and specificity under different α values in real data set.....237
Table 5.5 Sensitivity and specificity under different κ values 239
Table 5.6 Sensitivity and specificity under different k values 241

LIST OF FIGURES

Figure 1.1 Schematic representation of hemodynamic response function.....6
Figure 2.1 Symbolic representation of fMRI data with GLM 23
Figure 2.2 Hierarchical Clustering.....46
Figure 2.3. K-means clustering with spherical distributed data.....49
Figure 2.4 Failure of k-means with data in different manifolds 50
Figure 2.5 Spectral Clustering.....57
Figure 2.6 Distance from a point to a set. 64
Figure 2.7 Distance from a set to other set.....65
Figure 3.1 An example of real fMRI data 91
Figure 3.2 Estimated hemodynamic response function 92
Figure 3.3 Estimated convolution filter and noise process 93
Figure 3.4 Likelihood of the model.....95
Figure 3.5 The effect of the parameter κ on the algorithm when p=10 97
Figure 3.6 The effect of the parameter p on the convolution filter when κ=0.1 98
Figure 3.7 The effect of the parameter p on the algorithm when κ=10 99
Figure 3.8 Obtaining drift eliminated hemodynamic response.....104
Figure 3.9 Hausdorff distance between two sinusoidal signals 113
Figure 3.10 The effect of the time scaling parameter, τ, on the Hausdorff distance.....116
Figure 3.11 Hausdorff distance between two sinusoidal signals under zero mean unit variance additive independent Gaussian noises added to the signals above at every time instant.....118

Figure 3.12 Hausdorff distance between two sinusoidal signals under zero mean unit variance additive independent Gaussian noises added to the signal amplitudes ...119
Figure 3.13 Hausdorff distance between two sinusoidal signals under zero mean unit variance additive independent Gaussian noises added to the phase of signal D_2 ..120
Figure 3.14 Gaussian wavelet to simulate an outlier for a short interval of time 123
Figure 3.15 Two sinusoidal signals under additive Gaussian wavelet of short support as outliers.....124
Figure 3.16 Modified Hausdorff distance with different α parameters under outlier, no noise and Gaussian noise conditions.....125
Figure 3.17 fMRI signal with ID 608.....135
Figure 3.18 Cross correlation and modified Hausdorff distance between the fMRI with ID: 608 and its neighbors 136
Figure 3.19 Different neighbors of fMRI with ID 608 wrt modified Hausdorff and cross correlation metrics.....138
Figure 3.20 Comparison of the metrics.....144
Figure 4.1 fMRI signal generated using Block-Design Experiment.....152
Figure 4.2 Example of Event-Related Experiment 154
Figure 4.3 Hypothetical Hemodynamic Response Function.....157
Figure 4.4 An example of simulated fMRI and its stimulus pattern and its original BOLD response.....158
Figure 4.5 Estimated HRF, estimated stimulus and their convolution "Estimated BOLD Response" given a simulated fMRI signal with $\kappa = 10^{-5}$ and p=200 159
Figure 4.6 Estimated HRF, estimated stimulus and their convolution "Estimated BOLD Response" given a simulated fMRI signal with $\kappa = 10^{-4}$ and p=200.....161
Figure 4.7 Estimated HRF, estimated stimulus and their convolution, "Estimated BOLD Response", given a simulated fMRI signal with $\kappa = 10^{-3}$ and p=200 162
Figure 4.8 The real fMRI with ID_70 and its stimulus pattern.....163

Figure 4.9 Estimated HRF, estimated stimulus and their convolution, "Estimated BOLD Response", given the real fMRI signal with ID_70 164
Figure 4.10 Real fMRI data with ID_100 and its stimulus pattern 165
Figure 4.11 Estimated HRF, estimated stimulus and their convolution, "Estimated BOLD Response", given real fMRI signal with ID_100 166
Figure 4.12 Estimated HRF, estimated stimulus and their convolution, "Estimated BOLD Response", given real fMRI signal with ID_215 167
Figure 4.13 An example of real fMRI data from "*Data27*" with ID_11 168
Figure 4.14 Estimated HRF, estimated stimulus and their convolution, "Estimated BOLD Response", given real fMRI signal from "Data27"with ID_11 169
Figure 4.15 Modified HRF of the simulation data with σ_{AWGN}=4 171
Figure 4.16 Modified HRF of the simulation data with σ_{AWGN}=4, σ_{jitter}=4, σ_{lag}=16, σ_{drift}=16 172
Figure 4.17 Modified HRF of the real data with ID_6 173
Figure 4.18 Modified HRF of the real data with ID_135 174
Figure 4.19 fMRI Block Design Paradigm 176
Figure 4.20 a) fMRI Signal b) Hemodynamical Time Series obtained from 1st voxel of Data27 177
Figure 4.21 a) fMRI Signal b) Hemodynamical Time Series obtained from 4th voxel of Data27 178
Figure 4.22 a) fMRI Signal b) Hemodynamical Time Series obtained from 5th voxel of Data27 179
Figure 4.23 a) fMRI Signal b) Hemodynamical Time Series obtained from 16th voxel of Data27 180
Figure 4.24 a) fMRI Signal b) Hemodynamical Time Series obtained from 20th voxel of Data27 181
Figure 4.25 a) fMRI Signal b) Hemodynamical Time Series obtained from 27th voxel of Data27 182

Figure 4.26 fMRI Categorical Block Design Paradigm.....183
Figure 4.27 a) Applied stimuli b) fMRI Signal c) Hemodynamical Time Series obtained from 100th voxel of Active_male_1 185
Figure 4.28 a) Applied stimuli b) fMRI Signal c) Hemodynamical Time Series obtained from 135th voxel of Active_male_1 187
Figure 4.29 a) Applied stimuli b) fMRI Signal c) Hemodynamical Time Series obtained from 6th voxel of Active_male_1 189
Figure 4.30 a) Applied stimuli b) fMRI Signal c) Hemodynamical Time Series obtained from 18th voxel of Passive_male_1 191
Figure 4.31 a) Applied stimuli b) fMRI Signal c) Hemodynamical Time Series obtained from 60th voxel of Passive_male_1 193
Figure 4.32 a) Applied stimuli b) fMRI Signal c) Hemodynamical Time Series obtained from 140th voxel of Motion_male_1 195
Figure 4.33 a) Applied stimuli b) fMRI Signal c) Hemodynamical Time Series obtained from 40th voxel of Motion_male_1 197
Figure 4.34 Stimulus patterns for active and inactive voxel simulations 200
Figure 4.35 Examples of simulated active and passive BOLD signals.....201
Figure 4.36 Examples of simulated active and passive fMRI signals.....203
Figure 4.37 First 6 eigenvectors of Spectral Estimation 206
Figure 4.38 7.-12. Eigenvectors of Spectral Estimation 207
Figure 4.39 Eigenvalues of eigenvectors 208
Figure 4.40 Different distributions of the simulated data 209
Figure 4.41 Different clustering results on 2D plots.....210
Figure 4.42 ROC Curve for simulation data with $\sigma_{AWGN} = 4, \sigma_{jitter} = 4, \sigma_{lag} = 16, \sigma_{drift} = 16$ 212
Figure 4.43 Clustering on the raw fMRI data (without HRF extraction).....213
Figure 4.44 An example fMRI with AWGN noise of standard deviation 16, the hypothetical BOLD Response, and the estimated BOLD Response.....217

Figure 4.45 Clustering results with $\sigma_{drift} = 16$ 218

Figure 4.46 ROC Curve for simulation data with $\sigma_{drift} = 16$ 220

Figure 4.47 Eigenvalues when $\sigma_{lag} = 16,$ 221

Figure 4.48 Clustering of data into 17 clusters 222

Figure 4.49 First 4 eigenvectors of spectral estimation of real data 226

Figure 4.50 First 20 and 5 Eigenvalues of the eigenvectors respectively 227

Figure 4.51 Distribution of the real data using different eigenvectors 228

Figure 4.52 EM Clustering of the real data 229

Figure 5.1 The effect of the parameter *p* on the estimations of active and passive voxel responses 233

Figure 5.2 The effect of the parameter *κ* on the estimations of active and passive voxel responses 235

Figure 5.3 Clustering results of real data when α=2 238

Figure 5.4 Clustering of real data when α=10 238

Figure 5.5 Clustering results for simulation data when τ=1 240

Figure B.1 Comparison of extracted HRFs from simulated data with σ_{AWGN}=4 265

Figure B.2 Comparison of extracted HRFs from real data with voxel ID_100 266

Figure B.3 Comparison of extracted HRFs from real data with voxel ID_100 (ForWaRD uses the MAP estimation of stimulus pattern as input) 266

Figure B.4 Comparison of clustering results for simulated data 267

Figure B.5 Comparison of clustering results for real data 268

CHAPTER 1

1 INTRODUCTION

Researchers are heavily interested in identification of brain activation. Why do activations of the brain evoke such a wonder? Because the human brain is a barely explored new world with each activation producing a little sight of hidden functionality. Unfortunately, the current understanding of brain function is challenged with inconsistencies, errors and questions deserving answers.

Initial pioneering researches about brain activation were proposed in the nineteenth century. The idea of locating functional areas in the brain and mainly the smallest element of the functioning part is the present focus. In the succeeding decades, scientists were interested in the measurements of changes in brain physiology either caused by lesions or recorded as electrical pulses. The invasive nature of these measurements prevented a systematic study of the human brain. Nearly 200 years later, scientists are now attracted to **functional magnetic resonance imaging (fMRI)** and take the pictures of the active brain in both clinical and research settings. fMRI is a specialized type of

magnetic resonance imaging (MRI), which became popular among the neuroimaging society.

MRI is a technique for producing detailed images of the brain or other bodily structures, using a very strong magnet and radio waves. The subject lies on a table, with his head surrounded by a large magnet. The magnet causes particles inside the atoms, called protons, of the patient's brain to align with the static magnetic field. A pulse of radio waves is then directed at the patient's head and some of it is absorbed by the protons, knocking them out of alignment. The protons, however, gradually realign themselves, emitting radio waves as they do. These radio waves are captured by a radio receiver and are sent to a computer, which constructs the brain image by taking advantage of the different realignment timings of different tissue types.

fMRI uses a conventional MRI scanner, but takes advantage of two additional phenomena. The first is that blood contains iron, which is the oxygen-carrying part of hemoglobin inside red blood cells. Iron atoms cause small distortions in the magnetic field around them. The second phenomenon underlying fMRI is the physiological principle that whenever any part of the brain becomes active, the small blood vessels in that localized region become larger, causing more blood to flow. A large amount of oxygenated blood flows to the activated brain area, reducing the amount of oxygen-free (deoxy) hemoglobin. This causes a small change in the magnetic field, and thus in the fMRI signal, in the active region.

In order to understand the activations of the human brain, functional neuroimaging studies are needed which attempt to determine which areas are responsible for which mental processes. Although functional neuroimaging did not begin with fMRI, fMRI rapidly became one of the most popular and strong tools in neuroimaging. Before fMRI the most popular functional neuroimaging technique was positron emission tomography (PET) which is based on the injection of radioactive tracers that are attached to the

molecules to measure changes in the brain [1]. For example the radioactive isotope of fluorine can be attached to glucose, while the radioactive isotope of oxygen can be attached to the oxygen, then the distribution and changes in glucose and oxygen in the brain due to the sensory, motor and cognitive activity can be measured. However, the radiation used in the studies causes problems for human subjects who can take part in only a few PET scans. Also, generating the required isotopes is expensive for repetitive experiments. In addition, for obtaining a sufficient signal-to-noise (SNR) ratio, the data must be collected over a long period of time. Although PET imaging has several disadvantages, it still has critical importance in brain metabolism researches, since it directly measures the distribution of glucose or oxygen uptake in the brain by evaluating the number and timing of impact of the attached radioactive isotopes.

There are other techniques for studying brain functionality. The oldest approach is to investigate the effects of damage to the brain upon behavior. When there is damage in a brain area A that affects the behavior B, this approach concludes that the area A is necessary for the behavior B. However, due to the network structure of the brain, it can not be also concluded as the area A is sufficient for the behavior B. Also, the effects of damage usually changes over time and makes it difficult to track the effects of that damage. Furthermore, it is hard to find patients with isolated damages and the studies on animals can not be directly addressed to humans especially in the context of cognition processes. However, transcranial magnetic simulation (TMS) technique can interrupt functions temporarily within a brain region so that it can be used in human subjects [1]. TMS is a technique for temporarily stimulating a brain region to intervene with its function, enabling the identification of isolated damage affects on brain functionality. One limitation of TMS is that its range is mostly confined to cortex on the brain surface, restricting the range of areas to be studied with this paradigm.

Other methods for assessing brain function depend on drug manipulation. Neurons in the brain have receptors that are influenced by specific neurotransmitters. Drugs that affect

the action of these neurotransmitters possibly cause changes in the common areas of brain. However, systematic application of drugs is difficult to perform, and a number of drug manipulations need a long time period, in the order of a few weeks, to give results. Furthermore multiple regions with different functionality may be responsive to certain drugs, making identification of the activated regions difficult to detect.

Another major technique for brain function studies is based on the measurement of the electrical changes. Electrodes are inserted near or into a neuron, and the electrical potential is measured so that neural activity of that neuron can be measured directly. Since the electrodes cannot be inserted into neurons of the healthy human subjects, due to the invasiveness of the technique, electrical and magnetic activity of the brain can be measured outside the skull using electroencephalography (EEG) or magnetoencephalography (MEG) methods [1]. These methods are used for studying the timing of the brain processes since very rapid changes in electrical potentials and magnetic flux can be measured. However, it is impossible to uniquely detect the location of the neuron which causes the activity detected.

To sum up, functional neuroimaging is necessary to understand the brain activity and many methods are provided for this purpose. PET imaging suffers from invasiveness of the radioactive injections, the expense of obtaining radioactive isotopes and slow processing time. Studies depending on damages on the brain have pointed out that a brain region is dedicated to a certain behavior, but can not provide information about the timing of its activity or the specific function it is responsible for. Although, EEG and MEG give sufficient information about the timing of the activity, they do not provide clear information about the location of the responding neuronal populations.

Differing from other functional neuroimaging techniques, fMRI is a noninvasive technique so that it can be repeated as many times as needed using the same human subject. Also, fMRI localizes the brain activity more precisely (in mm resolution) to the

origin of the brain activity location. However, it is difficult, extensive and complex to analyze the fMRI data due to its low SNR ratio. Also, it is susceptible to several imaging artifacts as well as limited temporal resolution. Although it has these limitations, it is worth to employ fMRI for identifying brain activation and it has been rapidly provided as a primary investigate tool by thousands of researchers worldwide. Using fMRI, invaluable information about cognitive neuroscience, psychology, neurobiology, psychiatry, and radiology is provided.

fMRI is used to identify the brain activation by measuring the physiological activity that is correlated with neural activity. When there is a neural activation, the metabolic requirement (need for glucose and oxygen) and energy consumption of the neurons increase. To provide the energy needed, glucose and oxygen is supplied for the neurons. Oxygen is attached to hemoglobin molecules in the blood to reach the neurons that are consuming energy. Oxygenated hemoglobin (oxy-hemoglobin) is diamagnetic; that is, it has the property of a weak repulsion from a magnetic field. Deoxygenated hemoglobin (deoxy-hemoglobin) is paramagnetic; it has the property of being attached to a magnetic field. Neural activity increases energy demand and therefore increases oxygen consumption. Rushing oxygen into the active area washes out and reduces the existing deoxy-hemoglobin. Since deoxy-hemoglobin has paramagnetic properties, it has the effect of suppressing the MR signal, while oxy-hemoglobin does not. Cerebral blood flow refreshes areas of the brain that are active with oxygenated blood, lessening the local deoxy-hemoglobin, which in turn causes an increase in the measured MR signal in active brain regions. This is called **blood–oxygenation-level dependent (BOLD) fMRI** which is the most common measurement in neuroimaging. The increase in the fMRI signal triggered by neuronal energy consumption is known as the **hemodynamic response function (HRF)**. Figure 1.1 shows an illustration of a typical HRF. When the voxel is active, its HRF would look like a Gamma function as in Figure 1.1. The peak emerges as a response to the applied stimulus. In passive voxels the related HRF has no peaks and it has a flat shape since it does not respond to the given stimulus.

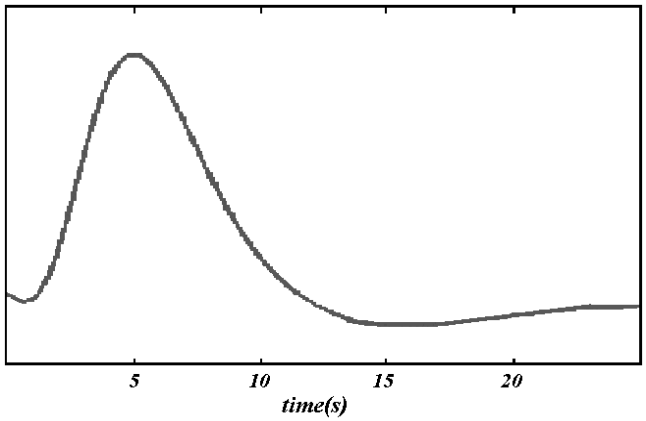


Figure 1.1 Schematic representation of hemodynamic response function

1.1 Thesis Objective

The main objective of this thesis is to identify brain activations so that given a voxel of the brain; we want to determine whether it is active or passive; and also to group these voxels according to their activation status due to a specific task.

fMRI is the most noninvasive technique among all other neuroimaging techniques and to get satisfactory results usually repetitive measurements are required. These measurements need to be performed on the same human subject for best performance and statistical reliability. Considerably short time periods are needed to obtain experimental results in the order minutes. So for the time being, fMRI is the best choice for brain activation analysis.

However, during fMRI experiments only one physical quantity is measured, which is the magnitude of the current in a detector coil measured by the MRI scanner. Unfortunately, this measurement which is composed of signal evoked from the neural activity is contaminated with noise. The noise arises from 1) intrinsic thermal noise within the subject and the scanner; 2) system noise due to the variations or discrepancies in the function of the scanner hardware; 3) head motion of the subject; 4) physiological factors including fluctuations in the blood flow, blood volume and oxygen metabolism; 5) changes in behavioral performance and cognitive strategy; and 6) variability in neural activity due to the non-task related brain processes. These unwanted variables in the fMRI data are tried to be removed by means of preprocessing processes (See Appendix A). The most insidious causes of noise are the head motion and physiological noise. Although there are some methods to correct the effects of these noise sources, they cannot be totally removed from the fMRI signal, which already presents a low SNR. So we need to extract the underlying hemodynamic response from the fMRI signal in order

to identify the brain activation since hemodynamic response is evoked predominantly due to neural activities.

Retrieval of the hemodynamic response is essential for a better understanding of neural activity. Since fMRI considers local changes in deoxy-hemoglobin concentration in the brain, hemodynamic response is needed to be obtained properly in order to interpret adequately the information provided by the fMRI signal to draw conclusions about the underlying neuronal activation. Once the hemodynamic response function (HRF) is obtained, analyzing its shape, the extracted features such as amplitude, delay, and duration is sufficient to infer information regarding the intensity, onset latency, and duration of the underlying brain metabolic activity. The onset and peak latencies of the hemodynamic response can provide information about the timing of activation for various brain areas and the width provides information about the duration of activation and cerebral dynamics [1]. Also, an appropriate hemodynamic response function leads to better statistical maps. With the low temporal resolution of fMRI signals, a binary baseline-activation description of the data is insufficient so it is necessary to take into account the temporal pattern of activation due to the hemodynamic response to the activation. Another reason is the possibility to give a physiological interpretation of the hemodynamic response function parameters and thus, to better understand the neurophysiology [3].

When functional relations between different voxels are considered, obtaining the hemodynamic response of the voxels separately may be inadequate. So given an fMRI data set, grouping them according to their similarity provides information for identifying sets of voxels whose activity both varies together over time and is maximally different from other sets. That is, due to a specific task, some voxels are active while the others are passive and grouping the hemodynamic responses into classes for identifying in each class, activity and passivity of voxels gives us their functional relationship. In addition, analyzing this relationship among the voxels has the potential of elucidating biological

relationships between the different functional areas of the brain. These hemodynamic responses are obtained for all fMRI data due to activities of given voxels by deconvolutions and are further classified to identify voxels which are active, passive, or exhibit other artifacts.

1.2 Methodology

There are several common methods for obtaining the hemodynamic response that leads to the understanding of how the application of certain stimuli leads to changes in neuronal activity. In this thesis we use a blind deconvolution technique to obtain the HRF. In the literature, researchers applied many other methods for the same purpose. But each has some limitations and drawbacks, which will be stated below. With the proposed blind deconvolution method we aimed to overcome these problems.

General Linear Model (GLM) approach has become the most known way to analyze fMRI data. It models the fMRI time series as a linear combination of several different signals. The stimulus function is known and the HRF is modeled using canonical HRF, typically either as a gamma function or the linear combinations of gamma functions [4], [5]. Since the shape of the HRF varies according to the brain area of subjects, assuming that the shape of the HRF is constant across all voxels and subjects as in GLM may give rise to misinterpretation of HRF. The assumption on the shape of the HRF is made more flexible by expressing the HRF as a linear combination of reference basis waveforms. One of the most flexible models, a finite impulse response (FIR) basis set, contains one free parameter for every time-point within a given window of time in every cognitive event [6], [7]. Also besides this, there are many other approaches that use different basis functions such as cosine functions [8], radial basis functions [9], spectral basis functions [10], Gaussian functions [11]. However, even if the used basis function differs in these

methods, a common problem for all of them is that they require an accurate match of the HRF with the combined basis functions. That means, in order to conclude that an estimated HRF belongs to an active voxel, the related fMRI has to match well enough to the idealized waveform of the HRF of an active voxel. Such approaches are known as **hypothesis-driven methods**.

Although hypothesis-driven analyses are used frequently, the model they try to fit to fMRI data does not always hold and the assumptions on the form of HRF may not always be valid. So, **data-driven methods** emerged which are not built on a priori statistical hypothesis and try to find the relationships among voxels over time. Two common data-driven methods are principal component analysis (PCA) [12] and independent component analysis (ICA) [13]. In PCA the orthogonal spatial patterns that capture greatest variance in the data are determined. The HRF is modeled linearly using these orthogonal components. However, if the hemodynamical changes are only a small part of the total signal variance; the obtained orthogonal components contain a little information about HRF and hence PCA may tend to model noise more accurately in comparison to HRF. Similarly, ICA determines the components of the data and the linear combinations of these components are used to model HRF. But the components are needed to be independent rather than orthogonal. The independent components are not ranked in order of appearance and it is therefore necessary to search for the important components to model HRF. Furthermore, both PCA and ICA assume that all voxels have similar statistical properties.

Estimating HRF out of the signal can be considered as a deconvolution problem since fMRI is assumed to be convolution of the neural activity evoking impulses and the underlying hemodynamic response. Moreover, artifacts like cardiac pulsation, scanner drift, habituation, subject motions also exist within the measured fMRI signal even though majority of these artifacts are eliminated during a preprocessing (See Appendix A). So, the hemodynamic response function can be extracted from the measured fMRI

time series in each voxel via deconvolution. We use blind deconvolution for this purpose. Since the objective of blind deconvolution is to reconstruct the original signal from a degraded measurement without the knowledge of either the original signal or the degradation process, it fits well to the fMRI problem. Because fMRI is a kind of degraded signal and it has many noisy components which have to be removed so that the original signal, HRF, due to the neural activity can be extracted. Furthermore, Blind deconvolution is a nonlinear process; therefore it is more suitable for HRF which has nonlinear characteristics. It is known that when two stimuli are presented successively, the resulting hemodynamic response is less than the sum of responses to the two stimuli separately [14]. So, linear approaches such as GLM are inadequate for many fMRI analyses. As a special case of blind deconvolution, we used Maximum A Posteriori (MAP) Blind Deconvolution in which we assume the HRF is convolved with an unknown convolution filter under some additive noise. This is expressed as a maximization problem over the probability distribution of the posterior using Bayes' rule.

However, all of the methods mentioned above, including blind deconvolution, are voxel-wise approaches in which the activations of the voxels are considered separately. They identify whether a specific given voxel exhibits significant task-related signal changes independently from other voxels. Nevertheless, when functional relations between different active voxels are considered, these approaches are inadequate since they are only specialized to determine the underlying neural activity of a specific voxel. For grouping activity responses of voxels together by their functional similarities which are HRF resemblances, many classification and clustering techniques are developed so that activated voxels can be separated from non-activated voxels. We aim at classifying extracted hemodynamics using their waveform features via MAP Blind Deconvolution to identify individual clusters as active voxels, passive voxels and other artifacts. In the future, a spatial component can also be added to extend our approach into voxel neighborhoods.

There are a number of different classification methods employed for fMRI data, each of which make different assumptions about the data and the training set used for learning the parameters of a model. Once the parameters are learned, the model is applied to predict the label of a previously unseen data set. Support vector machine (SVM) attempts to find a linear separating boundary in the feature space which is the original input space. Each data point in the input space represents a spatial pattern. In the event that the classes in the input space are not separable through a linear boundary, kernels are used for applying a non-linear transformation. In addition, the data that are least distinct from the members of other class are chosen as support vectors as they are at the boundaries of the classes [15]. But determining important aspects, such as features or statistical property of the training data is crucial for consistent separation of the test data arising from the same experiment. In linear discriminant analysis (LDA), one part of the data is used for training in order to distinguish between the population responses during dominance of either of the two percepts. Then the trained classifiers are applied to independently obtained test data to determine whether the time course of competing perception during their acquisition could be predicted. Also the data used for training and testing are independent time series collected at different times [16]. Logistic regression (LR) is another classification method applied to fMRI data. In LR the feature is selected by a developed method then with this selected feature, training of the model parameters is performed. Only a few parameters are selected as important, the other parameters are removed from the model. Different than SVM and LDA, the classification performance is not degraded with the increase of the number of irrelevant features in LR. But it suffers from computational burden.

All classification techniques require the label of each data in the training set to be given. In fMRI analysis it would be very difficult to use previously collected fMRI data sets as a training set since the fMRI signal depends on many factors such as experimental design, subject, and region of the brain activated. To overcome the need for a training set, clustering techniques have been developed.

In clustering analysis, researchers create similarity between the fMRI time series of different voxels so that the voxels can be grouped into distinct clusters, without the requirement for training data set. K-means is one of the popular clustering methods used with fMRI. It assigns each data to one of *k* groups, and iteratively determines the center of each group by calculating the average of its members [18]. However, k-means assumes that the data distribution is a mixture of *k* classes and the initial assignment of the number of clusters, *k*, is critical [19]. Also it forces the data into scatter with a hyper-spherical shape since the k-means algorithm implicitly assumes that the clusters in the data have diagonal covariance matrices (with the same variance values along the diagonal) resulting from the fact that, by design, it always splits the data halfway between the cluster means which is optimal only when clusters have (hyper-) spherical shape and so diagonal covariances. Hierarchical clustering is another applied method which initially considers each data as a member of its own cluster, and then the clusters are combined recursively according to developed similarity measure, eventually expecting every data to belong to a single cluster [20]. However, existence of the full hierarchy is not really appropriate for large fMRI data sets. In general fMRI signals have low SNR, which dramatically affects the quality of clusters generated by a hierarchical approach. This is because low SNR could result in incorrect grouping at early stages. Hierarchical clustering can never undo such errors since at every stage it operates on previously generated clusters. Also, some of the relationships exposed by hierarchical clustering may be only valid for a specific data set to which it is applied, and may not be generalized for other data sets. In addition, it tends to be very demanding computationally.

However, performing the clustering in the original high dimensional time domain using raw fMRI data usually suffer from the curse of dimensionality, which basically says that data spread out exponentially with the number of dimensions. This immediately results in sparsity and leads to low confidence in any statistical reasoning. Considering the fMRI data, we usually have the signals of dimensionality in the orders of hundreds

whereas the amount of existing data is not in exponential order of hundreds. The clustering would be simplified if the fMRI time series can be projected onto a lower dimensional space, so that clustering can be performed more easily. For this purpose, we used spectral clustering so that we can reduce the feature number, construct new features and improve the accuracy of the classifier.

The spectral clustering technique is formulated as a graph partitioning problem and the weight of each edge is the similarity between points that correspond to vertices connected by the edge. Spectral clustering tends to find the minimum weight cuts in the graph, and this problem can be addressed by the eigenvalue decomposition techniques, from which the term "spectral" is derived. The basis of the spectral clustering is the Laplacian of the similarity matrix, obtained from spectral graph partitioning. The success of spectral clustering is based on the fact that it does not make strong assumptions on the form of the clusters. As opposed to k-means or hierarchical clustering, where the resulting clusters form convex sets, spectral clustering can solve very general problems like intertwined spirals. Moreover, spectral clustering can be implemented efficiently even for large data sets like fMRI data sets. As all the clustering algorithms, spectral clustering can only be as good as the similarity metric used.

Euclidean distance, Mahalanobis distance, Gaussian similarity and Cosine similarity are the common metrics used in many clustering algorithms [21]. All of these metrics, as will be shown later in this work, is not convenient for fMRI data, since they are overly sensitive to shifts and delays in the signals. They are also sensitive to outliers such as some unusual activity of a short duration with respect to the general shape of a given signal. Therefore, we used Hausdorff distance metric for similarity measure.

The Hausdorff distance is a shape-comparison metric which performs well even when the image contains many features, multiple components, noise, spurious features, and occlusions. It matches two data without the requirement of point-to-point

correspondence, so the measure is insensitive to distortions. Since it is not based on point correspondence, it is tolerant to the delay in the two sets of features compared. However, when there exists an extraordinary noise peak in the data, this can make the Hausdorff distance unnecessarily large and in general very sensitive to outliers. In order to overcome this problem, we modified the Hausdorff distance to make it robust. The modified Hausdorff metric also allows and can tolerate small variations both in time and magnitude which is a big advantage as long as fMRI data is considered.

1.3 Contributions

In this thesis, our major contributions can be summarized as follows: (1) Seeing an fMRI signal as the convolution of the underlying neural impulses and the hemodynamic response function, we estimate the hemodynamic response function through MAP Blind deconvolution in completely model free settings, i.e, assuming no particular shape for the hemodynamic and no access to stimulus, secondly (2) In order to group the brain voxels, we use the output of blind deconvolution which includes the estimated hemodynamics as the input to Spectral Clustering using Hausdorff distance for a similarity measure. (3) Moreover, we present a thorough analysis of our proposed methods and algorithms for HRF estimation, clustering of obtained HRFs using the efficiency of Hausdorff distance.

Because the observed fMRI signals are seen as the convolution of the neural stimuli with the impulse function, so called hemodynamic response function, to estimate HRF usually a deconvolution technique is used. Most of the time, it is assumed that the stimulus is known or has a uniform effect on the brain voxels, and/or that a generic function or shape is assumed for HRF. Because of the very fact that HRFs and the stimulus reaching different regions of brain can vary between voxels, we take a

completely model free, unsupervised approach, blind deconvolution, to estimate HRF by using only fMRI signals. We do not incorporate any model for the unknown stimulus embedded in the fMRI signals and only use weak prior information regarding the HRF, such that an HRF should be smooth in a sense that the square sum of derivatives of that should be controlled.

Moreover, recent research shows that MAP based blind deconvolution techniques yields better results for the problem of image deblurring where the target image derivatives are thought to be sparse whereas for fMRI data analysis, hemodynamics should better be considered as having not sparse derivatives but small ones knowing that HRF ideally is a nicely behaving smooth function. Hence, we modify the MAP approach for HRF estimation accordingly and study the problem of HRF estimation within the Bayesian framework that provides the flexibility that one can treat the same problem by incorporating with different priors with the help of mathematical intuition that the Bayesian framework brings.

In order to group the similar brain voxels, for instance, into two groups: active voxels and passive voxels depending on that whether a voxel responds to a stimulus, fMRI clustering is required. Raw fMRI signals are known to suffer from low SNR. Instead of using raw signals, the HRF is estimated beforehands, as described above for further clustering them into voxel groups. The source of the separation between an active voxel and a passive voxels is that usually an fMRI signal recorded for an active voxels reflects a strong correlation with the stimulus. On the other hand, fMRI signals for passive voxels mostly behave randomly. Hence, one can easily expect a baseline shape for active fMRIs correlated with the stimulus, and no particular shape for passive fMRIs. Usually fMRI signals are checked with the stimulus via cross correlations and then based on that, activation is declared if the correlations are large enough. This approach also assumes that stimulus is known. Moreover, when there is a weak correlation with the stimulus for a given fMRI signal it can still belong to an active voxel if it can be coupled with other

active voxels or physically placed close to a set of active pixels. Hence, declaring voxel based activation might result in low activation detection. In this thesis, we analyze voxels in their clusters, not independently.

As it is true for all type of data analyses including clustering, choosing a right distance/similarity metric is very important. In general, Hausdorff distance is widely used for shape matching in computer vision problems. We propose to use it in fMRI clustering to detect the similar hemodynamic shapes to form the class of active voxels. To the best of our knowledge, Hausdorff distance has not been used for this purpose before and the modification we provide for this distance metric is novel. We analyze its properties and efficacy when it is used under the topology of fMRI data. In clustering problems, often feature extraction is used in order to extract the relevant piece of information hidden in the data. In this thesis, with the help of Hausdorff distance, we do not conduct any feature extraction and so avoid the burden of such an extra process.

1.4 Outline of the Thesis

In Chapter 2, recent works in the literature for HRF extraction, their advantages and limitations are stated. Then mathematical background is provided through the definition and mathematical explanation of blind deconvolution. The most known blind deconvolution methods are also presented together with their applications in the literature. Then Maximum A Posteriori (MAP) Blind Deconvolution method is defined and why it is suitable to fMRI problem is discussed in details.

In Chapter 3 we introduce our methodology. First the mathematical definition of MAP Blind Deconvolution is expressed. During implementation, we make a few assumptions on HRF and convolution filter. These assumptions and their validity are also discussed.

In the same chapter, the meanings of the parameters in the algorithm are demonstrated using examples. Then, the spectral clustering used for classifying hemodynamics in our approach is then given. There, the chosen distance measure, which is the Hausdorff distance, is discussed in details and it is compared with other common metrics showing the superiority of Hausdorff distance.

In Chapter 4, experimental results of our approach are discussed based on simulation datasets first. Adding different noises (drift, lag, jitter and white noise) to the simulated data, robustness of our algorithm is tested. Then the same experiments are performed on real fMRI data sets. In the experiments, first HRF extraction results are given and discussed. After that, hemodynamical time series are extracted from real fMRI data sets; their neural and physiological meanings are elaborated. These time series become the input of the proposed clustering algorithm, enabling the spectral clustering of these time series under the modified Hausdorff distance. The results are discussed in both simulated and real data sets.

In Chapter 5, sensitivity analysis based on our simulation as well as the real data is conducted on our algorithm. Through this analysis, the parameters of our algorithm are optimized, after detailed discussions on the effect of each parameter on the deconvolution and clustering performances. The sensitivity and robustness of the algorithm are then demonstrated for ranges of system parameters declared optimum.

In Chapter 6 several conclusions are drawn from the results obtained in this thesis. Finally some directions for further research are recommended.

CHAPTER 2

2 LITERATURE SURVEY AND MATHEMATICAL BACKGROUND

A broad literature exists on identification of neural activity of the brain from several different perspectives. One perspective is to localize regions of the brain, activated by certain task. Others focus on determining distributed networks that correspond to brain functions and thus predicting possible psychiatric and disease populations. Regardless of what the perspective of the analysis is, all of the researches are trying to understand how presentation of a certain stimuli causes changes within the brain.

The analysis of fMRI has been undertaken using many different approaches. Some of them analyze the data voxel-by-voxel considering all the voxels independently. These approaches also differ from each other according the assumptions they are based upon. Among these methods a number of approaches match a model to the fMRI data, which are known as **hypothesis-driven methods**. Other researchers use fMRI data directly to explore the structure of the data expecting that task-related activations will emerge from the process. Such approaches are known as **data-driven methods**.

When the relationship between the voxels is considered, the relation pattern should be grouped yielding classes of voxels that can be labeled as active or passive under the administered stimuli. Based on this, researchers provided many **classification** and **clustering techniques**.

Our purpose in this thesis is to identify brain activations using the measured fMRI signal and classify each voxel according to the relation between their underlying hemodynamic responses. Different than the previous works in literature, we use a combination of a data driven-method, namely **blind deconvolution**, and a clustering technique, spectral clustering, in our approach. fMRI analysis can be considered as a deconvolution problem since fMR time series is the convolution of the input stimuli and the underlying hemodynamic response. So, the hemodynamic response function can be extracted from the measured fMRI via deconvolution. We use blind deconvolution for this purpose. Since the objective of blind deconvolution is to reconstruct the original signal from a degraded measurement without the knowledge of either the original signal or the degradation process, it suits the fMRI problem well. For this purpose we used Maximum A Posteriori (MAP) Blind Deconvolution which assumes smoothness of the HRF. Then to increase the smoothness of the HRF we maximize the posterior distribution of it using Bayes' rule. By means of MAP Blind Deconvolution, we estimate the HRF of each independent voxel and the underlying stimulus pattern given only the fMRI data. In this chapter we will give in details, definition and applications of blind deconvolution in literature; and clarify why blind deconvolution is suitable to the fMRI analysis. In the following chapters we will explain how we adapted MAP Blind Deconvolution to our fMRI analysis.

After obtaining the hemodynamic response of each individual voxel, we classify them according to their hemodynamical relation pattern via **spectral clustering** to detect whether the voxels are active or passive. fMRI data is high-dimensional and spectral clustering can transform the data to a proper feature space, greatly reducing the number

of features without losing much information. So, spectral clustering is suitable for fMRI data classification. In this chapter we will explain the spectral clustering algorithm and its applications in details.

The clustering techniques can only be as good as the distance measure used. There are several similarity measures like Euclidean, Mahalanobis distances, Gaussian and Cosine similarities which can not deal with the shifts, drifts and outliers among the signals. Hausdorff distance outperforms these metrics in the fMRI analysis which will be explained with examples in the next chapter. In this chapter in order to establish familiarity with the Hausdorff distance, the definition and applications of Hausdorff distance in literature will be explained.

To sum up, in this chapter, previous studies on fMRI, their limitations and drawbacks will be given. Also, how-well these analyses fit to fMRI data will be discussed. Moreover, as the basic approach of this thesis, deconvolution techniques will be overviewed and in particular common blind deconvolution techniques and our MAP Blind Deconvolution will be introduced. Classification techniques will be analyzed, from those existing in the literature with a special emphasis given to spectral clustering. Common distance measures used in clustering will be discussed and why we chose the Hausdorff distance among those will be clarified.

2.1 Approaches to FMRI Analysis

2.1.1 Hypothesis-Driven Methods

General linear model (GLM), as its name implies, represents the given data in a linear model as;

$$y = a_0 + a_1 x_1 + a_2 x_2 + ...a_n x_n + e \tag{2.1}$$

The observed data, *y*, is equal to a weighted combination of the system state variables, x_i, plus an additive noise, *e,* which is assumed to be independent and identically distributed normal random variables with zero mean and variance σ^2. The parameter weights, a_i, expresses the importance of the state variables. The term a_0 represents the total DC component.

For fMRI analysis, there is only one known quantity, which is fMRI data. The model is made to fit the data so that the weighted sum of the system states produces the closest match to the actual data time series. In Figure 2.1 a number of state variables are estimated according to what the response should look like to each type of experimental stimulus. Then the best parameter weights, $\alpha_{i\ s}$ are calculated to match the given fMRI data. However, in GLM only one set of model factors is used for every voxel in the analysis.

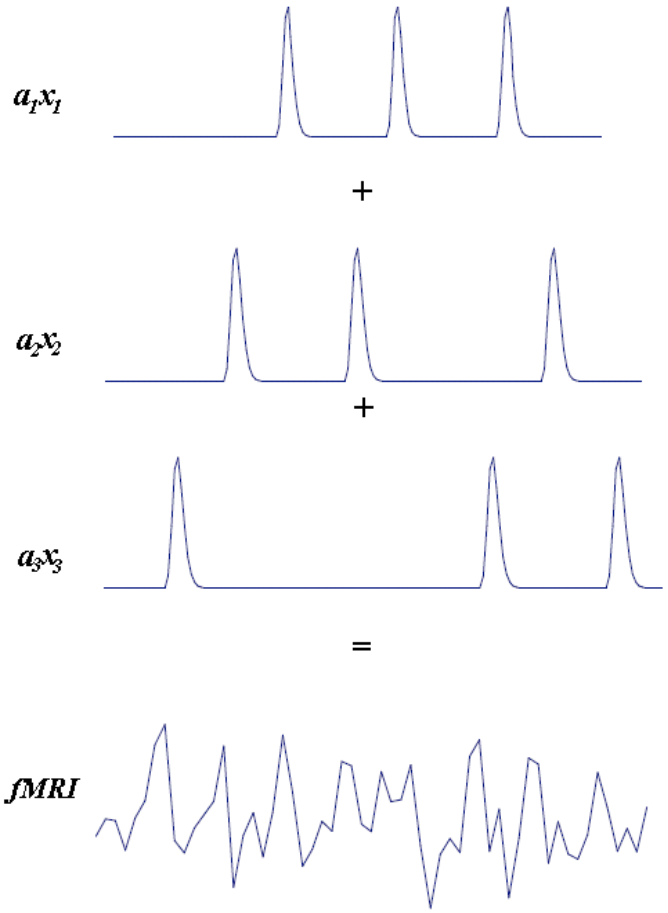


Figure 2.1 Symbolic representation of fMRI data with GLM

(2.1) can be represented in a matrix form as;

$$\mathbf{Y} = \mathbf{XA} + \mathbf{E} \tag{2.2}$$

where **Y** is a matrix with series of multivariate measurements, **X** is the design matrix that is estimated by the experimenter, **A** is the parameter matrix that is calculated during the experiments and **E** is a matrix containing errors or noise.

The success of a general linear model solely depends on the validity of the design matrix model. The design matrix consists of one or more state variables that represent possible contributors to the fMRI time series.

There are several assumptions that cause many limitations in GLM. First of all, the choice of the design matrix is crucial for the validity of the analysis. One assumption is the use of the same design matrix throughout all voxels. A model factor that is correct for one region may be incorrect for another since the dynamics of the hemodynamic response differ from region to region. Also, the amount of the noise in a voxel does not depend on the task condition according to GLM. However, the noise levels are higher during the activation than during the rest. Another assumption is that all of the voxels are considered independently. It is expected by the experts that adjacent voxels tend to have very similar properties. Moreover, GLM assumes that each time point is independent from all others.

Although, GLM is limited because of these assumptions, many researchers are attracted by its simplicity and satisfactory results. All the researches impose the importance of the design matrix construction. In [25], design matrix is divided into two partitions in order to separate the interest effects and confounding effects to obtain reliable estimates of GLM coefficients. Confounding here means an uninteresting effect that could defeat the estimation of interesting effects, i.e, the confounding effect of global changes on regional activations. However, for proper division of design matrix, gathering several fMRI data is required. In [26], the basis functions in the design matrix are chosen as exponentially modulated sine functions which resemble typical of response profiles seen in the fMRI so that modeling HRF by means of these basis functions become reasonable. In the traditional GLM approach the fMRI data is analyzed imposing an estimated model to the hemodynamic response. However, mismatches between this estimation and the underlying hemodynamic response of the fMRI data can be present between voxels, slice-timing differences, and also between groups of subjects. The use of a

hemodynamic model and its temporal derivative for fMRI analysis add flexibility for delay-induced modeling mismatches. The effects modeled by the derivative terms are interpreted as a shift of the hemodynamic model in time. Using these derivatives, minimization of effects of delay in hemodynamic response is provided. [27].

The fMRI signal can also be modeled with an FIR filter in order not to impose a specific shape to the hemodynamic response function since HRF has non-systematic behavior due to the applied stimuli. FIR models the fMRI signal by a linear combination of coefficients multiplied by the stimuli pattern. These coefficients are estimated by a maximum likelihood (ML) solution. The filter is needed to be smooth because as the number of parameters increases, there is a risk that the model will overfit or parameters become ill determined. To overcome this problem a Gaussian process prior on the filter parameters is introduced. A drawback of this approach is due to the computational complexities. Therefore one of the biggest challenges of this approach lies in the practical implementation for whole brain analysis, or for the analysis of a reasonable subset of voxels [3]. The idea of the smoothness matrix is not applicable when hemodynamics depends both on spatial location and on stimulation conditions. Gaussian functions may also be used for modeling hemodynamic response. The fMRI signal is considered as the linear convolution of the stimulus function and the Gaussian-shaped hemodynamic response function. A candidate model for HRF is said to have enough degrees of freedom to fit the actual hemodynamic response, and its parameters should highlight significant aspects of related biological event. So, Gaussian functions may be chosen for modeling the HRF. But it is not sufficient to make definitive conclusions about the nature of the variation of hemodynamic parameters, which depend on a lot of factors such as brain region, stimulus condition and frequency, or cognitive processes unique to each individual subject [11].

In the Bayesian analysis, the fMRI data is modeled using a mixture of two Gaussian distributions. This mixture aims to discriminate active areas from passive ones in the

region of interest. Mean value of one of the two Gaussian distributions is zero which represents the mean of the amplitude levels for passive voxels. The hemodynamic response is assumed to be smooth enough in the time domain, which is assured by the prior Gaussian estimation [28]. Bayesian approach is also used for modeling the noise in the fMRI data. Since the noisy nature of the fMRI data makes the detection of the hemodynamic response difficult and challenging, the estimation of the noise is important in fMRI analysis. In the Bayesian approach, the noise term is modeled as nonstationary, since the fMRI noises are inherently time varying due to the movements of the subject, or the warming of the scanner coils during rapid switching for fast data acquisition. The Bayesian estimator estimates the weights in general linear model for the fMRI noise model [29].

In **State-Space Model**, the hemodynamic response is formulated into a state-space model by applying the Local Linearization (LL) methodology by which the parameters and the underlying states of the system generating responses can be estimated. These states are the dynamics of the system which are a flow-inducing signal triggered by neuronal activation, deoxyhemoglobin, cerebral blood flow and volume. This model is used to simulate the trajectories of the state-space variables, and these four state variables are used to model the observed fMRI signal. In the context of the LL filter theory, the equations for the evolution of the conditional mean and the covariance matrix of the state-space variables are properly defined for the hemodynamic approach. Equations for differences, which depend on the LL filter gain, are also reported and during the estimation procedure, they are only evaluated on those time instants where data is available, as defined in the LL filtering strategy for the case of missing values. Additionally, radial basis functions have been introduced as a parametric model to represent arbitrary temporal input sequences in the hemodynamic approach, which could be essential to understanding those brain areas indirectly related to the stimulus. However, it is not a practical method due to the large computational costs involved to determine the brain activation areas. Furthermore, this method assumes that the

uncertainty of the model originating solely from instrumental errors and ignores physiological effects [30].

As explained above, in hypothesis-driven methods, a model of the expected hemodynamic response is generated and compared with the data. These methods require prior knowledge of event timing from which an expected hemodynamic response can be modeled. Although accurate experimental paradigms may be available, thorough understanding of the hemodynamic changes that relate neuronal activity to the measured fMRI signal is still not defined accurately. In addition, they are not adaptive to the canonical HRF. The canonical HRF does not fully consider subject-wise differences and experimental variance or unpredicted phenomena during the task period, thereby reducing the sensitivity of detection in hypothesis-driven methods. Also, for brain responses that are not directly locked to the paradigm, model-driven analysis may not be adequate.

2.1.2 Data-Driven Methods

2.1.2.1 Principal Component Analysis (PCA)

Principal component analysis (PCA) is a classical statistical method that transforms a correlated variable set into a set of uncorrelated variables, which are called principal components. It reduces data dimensionality by performing a covariance analysis among data set. This linear transform has been widely used in data analysis. Principal component analysis is based on the statistical representation of a random variable. Given a data set, **x**, where

$$x = (x_1, ..., x_n)^T \tag{2.3}$$

and the mean of that data set is denoted by

$$\mu_X = E\{x\} \tag{2.4}$$

and the covariance matrix of the same data set is

$$C_x = E\{(x - \mu_x)(x - \mu_x)^T\} \tag{2.5}$$

The components of C_x, denoted by c_{ij}, represent the covariances between the random variable components x_i and x_j. The component c_{ii} is the variance of the component x_i. The variance of a component indicates the spread of the component values around its mean value. If two components x_i and x_j of the data are uncorrelated, their covariance is zero ($c_{ij}=c_{ji}=0$). The covariance matrix is, by definition, always symmetric.

From a sample of vectors $x_1,\ldots, x_m$, the sample mean and the sample covariance matrix can be calculated as the estimates of the mean and the covariance matrix.

From the covariance matrix, an orthogonal basis by finding its eigenvalues and eigenvectors can be calculated. These are important, as they give useful information to characterize the concerned data in fewer dimensions. The eigenvectors e_i and the corresponding eigenvalues λ_i are the solutions of the equation

$$C_x e_i = \lambda_i e_i, \qquad i = 1, ..., n \tag{2.6}$$

The λ_is are assumed to be distinct. These values can be found, for example, by finding the solutions of the characteristic equation:

$$\left|\mathbf{C}_x - \lambda\mathbf{I}\right| = 0 \tag{2.7}$$

where the **I** is the identity matrix having the same order with Cx and the |.| denotes the determinant of the matrix. If the data vector has *n* components, the characteristic equation becomes of order *n*. This is easy to solve only if *n* is small. Solving eigenvalues and corresponding eigenvectors is a non-trivial task, and many methods exist. By ordering the eigenvectors in the order of descending eigenvalues, one can create an ordered orthogonal basis with the first eigenvector having the direction of largest variance of the data. In this way, directions in which the data set has the most significant amounts of energy can be found.

Suppose a data set is given of which the sample mean and the covariance matrix have been calculated. Let **A** be a matrix consisting of eigenvectors of the covariance matrix as the row vectors.
By transforming a data vector **x**, it is obtained that;

$$\mathbf{y} = \mathbf{A}(\mathbf{x} - \boldsymbol{\mu}_x) \tag{2.8}$$

which is a point in the orthogonal coordinate system defined by the eigenvectors. Components of **y** can be seen as the coordinates in the orthogonal base.

As an alternative approach, instead of using all the eigenvectors of the covariance matrix, the data can be represented in terms of only a few basis vectors of the orthogonal basis. The matrix having the first *k* eigenvectors can be represented as rows by $\mathbf{A_k}$, a similar transformation can be obtained as;

$$\mathbf{y} = \mathbf{A}_K(\mathbf{x} - \boldsymbol{\mu}_x) \tag{2.9}$$

This means that the original data vector is projected on the coordinate axes having the dimension *k*. This inner product minimizes the mean-square error between the data and the representation based on a given number of eigenvectors.

If the data is concentrated in a linear subspace, this provides a way to compress data without losing much information, thus simplifying the representation. By picking the eigenvectors having the largest eigenvalues we lose as little information as possible in the mean-square sense.

When applied to fMRI, PCA runs into serious difficulties since the data have extremely high dimensionality relative to the number of observations. The covariance matrix on which the analysis is performed is sometimes a poor estimate of the real data covariance. Because of this, using PCA directly in the fMRI vector space might suffer from curse of dimensionality. So in most studies, PCA is limited to predetermined small regions of brain. This means that the PCA is carried out mostly in a subspace where the experimental regressors dominate, which will be much smaller than the entire space of the original data. Although this is a meaningful way of employing PCA, it also limits its scope in exploratory analyses, because the signal of interest has already been identified by other means. Instead, one can see fMRI time series as continuous functions of time sampled at the interscan interval and subject to observational noise. These functions may be estimated by fitting a set of basis functions to each voxel time series. Collectively, the functions replace the voxels of a series of images with a single “functional image.”

Then, the eigenanalysis is carried out directly on these functions. Depending on the assumptions which the hemodynamic response is restricted to, this set of basis function can be designed many ways. Hence, the hemodynamic response for each voxel can be calculated by projecting its estimated function onto the principal components [31].

2.1.2.2 Independent Component Analysis (ICA)

Independent component analysis (ICA) is a statistical and computational technique for revealing hidden factors that underlie sets of random variables, measurements, or signals. It regenerates the observed data, which is typically given as a large database of samples. Since fMRI data sets contain mixtures of many different sources of variability including physiological signals, fluctuations due to head motion and noise and also the hemodynamics, ICA is used to separate fMRI data into spatially independent patterns of activity.

In a linear decomposition of fMRI data, the data matrix can be transformed into a set of volume maps, $\boldsymbol{C}$, by taking linear combinations, defined by an n by n matrix $\boldsymbol{W}$, of the volumes recorded at each time point

$$\mathbf{C} = \mathbf{WX} \tag{2.10}$$

where $\boldsymbol{X}$ is an n x v row mean-zero data matrix (where n is the number of time points in the experiment and v is the number of brain voxels), $\boldsymbol{W}$ is an n x n matrix containing combinations of volumes and $\boldsymbol{C}$ is an n x v matrix of component maps. $\boldsymbol{W}$ is selected so that the resultant component maps $\boldsymbol{C}$ are uncorrelated and summarize the variability in

the data in as few maps as possible. ICA is a generalization of PCA that selects $\boldsymbol{W}$ so that the rows in $\boldsymbol{C}$ are independent.

ICA assumes that the fMRI data are composed of the linear sum of either spatially or temporally independent patterns of activity. In ICA, there is no explicit noise model; rather, the noise was assumed to be distributed among one or more of the components. The components are assumed to be fixed throughout the fMRI experiment and the relative contribution from each component at a given time point in the experiment is the same throughout the brain A weighted sum of the extracted components is used to model the individual time series at each voxel. ICA does not require a reference function. In principle, it identifies the pure time series resulting from the convolution of the hemodynamic responses with the stimuli, in addition to a multitude of other noise-related components such as drift, motion, and flow artifacts. However, ICA has not been optimized for fMRI studies. The effects of data matrix size, filtering, and task paradigm on the sensitivity and specificity of ICA are unknown. One other weakness of ICA is that the trends in the data may be fragmented into multiple components, each with a highly related time course [10].

Besides these assumptions for fMRI analysis, ICA also has some ambiguities that are valid for all ICA applications. These are; the variances or the energies of the independent components cannot be determined and also the order of the independent components is ambiguous [34].

To sum up, data-driven methods examine the fMRI data statistically without any assumption about the paradigm or the hemodynamic response function. They can naturally provide an alternative to comparing each voxel's time points against a model hypothesis. They explore the data to find "interesting" components or underlying sources which is HRF in our problem. For example, structures or patterns in the data, which are difficult to identify a priori, are motion related artifacts, and drifts. The

flexibility of analyzing the fMRI signal without any constraints on the model is inspiring especially where it is difficult to generate a good model. However, among the data-driven methods, there are drawbacks in PCA and ICA. For example, the assumption implicit in PCA is that all components are Gaussian and uncorrelated whereas ICA assumes that all components are non-Gaussian and independent. In addition, a significant estimate for each component is usually not available. Also, these methods do not provide statistics for inferences about whether a component varies over time and when changes occur in the time series. In addition, because they do not contain any model information, they capture regularities whatever the source is; thus, they are highly susceptible to noise, and components can be dominated by artifacts.

Besides, hemodynamic response functions vary from subject to subject, from region to region and from task to task, so there is not a stationary and linear straightforward correspondence between the fMRI signal and the underlying hemodynamic response function. Also, hemodynamic changes result from a complex nonlinear dynamical system, which is neuronal synaptic activity, so the fMRI signals have nonlinear characteristics. Moreover, the low signal to noise ratio of the fMRI signals makes the problem more complicated. A strong limitation that arises in the above-mentioned methods is their lack of robustness, mainly due to these reasons. Blind deconvolution seems to overcome the problems that the other methods faced with.

2.1.3 Blind Deconvolution Applications

In general, the relationship between initial neuronal activation and the observed fMRI rests on a complex physiological process. If this process is known and well described, it can be approximated by mathematical modeling. However this process is still not known well enough in order to have a powerful model. And also assuming a global model

across all voxels of brain is not desired. On the other hand, deconvolution is a good candidate to take a rather general approach instead of imposing a model in the hemodynamic response estimation process, which assumes fMRI as convolution of the underlying HRF with experimental stimuli that can evoke different response for every voxel. Since we aim at unveiling the HRF, which is buried within a convolution, a deconvolution technique is required. When neither of the convolved signals are known, then the deconvolution is named as 'blind'. Blind deconvolution is a model free approach and has a great generality in terms of the capability of generating a rich family of time series signals. Mathematically;

$$r(t) = d(t) \otimes k(t) + n(t) \tag{2.11}$$

where

r(t) : observed signal

d(t) : desired original signal

k(t) : blurring function

n(t): noise

Although convolution is usually defined as a linear operation, the term deconvolution is generally used in reference to the inversion of nonlinear convolution models. Considered as a challenging problem, the blind deconvolution has been the topic of such numerous research efforts and various blind deconvolution methods have been proposed.

2.1.3.1 Blind Deconvolution Methods

There are a few methods widely used in blind deconvolution, including Priori Blur Identification Methods, Zero Sheet Separation, Auto Regressive Moving Average (ARMA) Estimate and Nonparametric Iterative Methods. This section will describe these methods in general.

In these blind deconvolution methods preprocessing is needed before deconvolution process is performed to eliminate the noise term, *n(t)* given in (1). These preprocessing processes will be mentioned after the blind deconvolution methods are stated. So, in most of these methods the noise term will be ignored.

2.1.3.1.1 Priori Blur Identification Methods

The a priori blur identification methods are the class of methods that perform the blind deconvolution by identifying the blurring function prior to the restoration. Typically, these methods assume the blurring function to be of a known parametric form. The associated parameters are unknown and to be determined before restoration. The a priori blur identification methods are relatively simple to implement and require low computational complexity. But they have some major drawbacks such that they require the knowledge of the form of the blurring function and also additive noise can mask the blurring function and thus degrade the performance [65].

2.1.3.1.2 Zero Sheet Separation Method

The method of zero sheet separation provides valuable insight into the blind deconvolution problem in multiple dimensions. It depends on the analytical properties of Z-Transform. After taking the Z-Transform of the degraded image, the deconvolution problem will be solvable if the Z-Transform of the degraded image can be factorized. The zero sheet separation algorithms are such a factorization scheme. However, it suffers from several major problems. First, the method is highly sensitive to noise. Second, it has computational complexity and finally, the method is prone to inaccuracy for larger images [65].

2.1.3.1.3 Auto Regressive Moving Average (ARMA) Estimate

This method regards the original image as a 2 dimensional AutoRegressive (AR) process and blurring function as a 2 dimensional Moving Average (MA). The blurred image is therefore modeled as the autoregressive moving average (ARMA) process. Therefore, blind deconvolution is translated into the problem of determining the parameter of ARMA.

There are several algorithms to estimate the ARMA parameters, including Maximum Likelihood, Generalized Cross Validation (GCV), Neural Network and High Order Statistics (HOS) etc. They all have good robustness on noise, but when there are too many parameters, they cannot convergent to global optimality [66].

2.1.3.1.4 Nonparametric Iterative Methods

This class of methods does not require certain parametric form for the true image or the Point Spread Function (PSF). They are therefore referred to as the nonparametric methods. One common feature of these methods is that they all assume certain constraints on the original image and the blurring function. There exist different algorithms that belong to this category, including nonnegativity and support constraints recursive inverse filtering (NAS-RIF) and iterative blind deconvolution (IBD) method.

2.1.3.1.4.1 NAS-RIF Method

The nonnegativity and support constraints recursive inverse filtering (NAS-RIF) method is an iterative algorithm where the coefficients of the inverse filter are updated iteratively to minimize a convex cost function. Consider the convolution system represented by (2.11), the algorithm employs an inverse filter g(t) to convolve with the blurred image r(t) to obtain an estimate of the original image $\hat{d}(t)$;

$$\hat{d}(t) = r(t) \otimes g(t) \tag{2.12}$$

Once $\hat{d}(t)$ is obtained, the known characteristics are imposed onto the original signal to update it and the updated signal is denoted as $\hat{d}_u(t)$. The cost function J is then defined as the difference between the estimation $\hat{d}(t)$ and the projected estimation $\hat{d}_u(t)$:

$$J = \left\| \hat{d}_u(t) - \hat{d}(t) \right\|^2 \tag{2.13}$$

The aim is to minimize the cost function, so a number of iterations are to be performed. The advantage of the technique is that convergence to a feasible set solution is guaranteed, while a lot of other existing approaches do not guarantee convergent solutions. The disadvantage of NAS-RIF is that it is sensitive to noise [67].

2.1.3.1.4.2 Iterative Blind Deconvolution (IBD) Method

In the IBD method, first, a nonnegative initial guess is made for the original signal. After a random initial guess is made for the original signal, the signal is Fourier transformed which is then inverted to form an inverse filter and multiplied by the observed signal to form a first estimate of the blurring function's spectrum. The algorithm alternates between the signal and Fourier domains, enforcing known constraints in each signal. The constraints are based upon information available about the original signal and blurring function. The iterative loop is repeated until two positive functions have been found [65]. However, the inverse filter associates problems in the regions of the signal with low value. Also, the length of the estimated convolution components cannot be controlled since the basis functions of Fourier Transform is periodic so the length of each of them is infinite.

Sometimes we can pose a priori information on the signals we want to estimate. Such information can come either from the correct knowledge about the physical process or from previous empirical evidence. We can impose such prior information in terms of a probability distribution function (PDF) on the signal to be estimated. These probabilities are called the prior probabilities. We refer to the inference based on such priors as Bayesian inference. Bayes' theorem shows the way for incorporating prior information in the estimation process:

$$p(\theta \mid x) = \frac{p(x \mid \theta) p(\theta)}{p(x)} \tag{2.14}$$

The term $p(\theta \mid x)$ is called the posterior. $p(x \mid \theta)$ is the likelihood term and $p(\theta)$ is the prior term. *p(x)* serves as a normalization term so that the posterior PDF integrates to unity. Thus, Bayesian inference produces the maximum a posteriori (MAP) estimate.

$$\arg\max_{\theta} p(\theta \mid x) = \arg\max_{\theta} p(x \mid \theta) p(\theta) \tag{2.15}$$

This is the most general approach in MAP estimation. The method is based on the prior assumptions about the specific problem and develops in accordance with the problem characteristics. In our fMRI problem the priors are about the smoothness of the hemodynamic response function and also the positivity of the blurring signal. We expect the estimated HRF to be smooth and the other convolved signal (to which we will call convolution filter later) be finite and positive. The details of the method will be clarified step by step in the next chapter considering the fMRI problem.

In the literature there are a few blind deconvolution applications using fMRI data. In [35], a semi-blind approach is applied. Since the stimuli timing is being known, structural knowledge on the input sequence is available to constrain the HRF estimation, which leads to a semi-blind deconvolution problem. The HRF is modeled as a smooth

function. The hemodynamic levels are assumed to be independent across voxels as well as across stimulus types or conditions. To estimate the HRF model parameters, a Bayesian approach is developed using physiological prior information. The parameters that represent the noise variances of voxels and also location and scaling parameters are iteratively estimated using Gibbs sampler algorithm and this estimation is the semi-blind deconvolution part of the algorithm. Gibbs sampling starts with a seed parameter vector and sequentially modifies one component at a time by sampling according to the full conditional probability distribution function of that component given the remaining variables and the data. The parameters are updated with a satisfactory number of iterations. The approach in the aforementioned reference is tested only with synthetic data.

Another blind deconvolution algorithm is applied to an fMRI data set to separate it into a HRF and an unknown series of presentation times of stimuli [36]. The values of the cost function are used for this separation algorithm as measures of a neural activity. The cost function consists of two terms. One is an error term representing discrepancy from a conventional convolution model of fMRI. The other term represents statistical characteristics of the estimated stimuli presentation time series. In this approach, only one waveform of HRF is assumed for one voxel and one voxel is activated by only one kind of stimulus or task. Also the regions activated and deactivated by presentations of objects to the patient are determined using a software program before the proposed algorithm is applied [36].

Blind deconvolution is also applied to voxel-specific arterial input functions in dynamic contrast brain perfusion imaging. Voxel-specific arterial input functions are estimated blindly using the theory of homomorphic transformations and complex cepstrum analysis. Wiener filtering is used in the subsequent deconvolution. Tissue residue function is the blurring function. In the cepstrum domain there is a cut-off frequency to separate the arterial input function and the tissue residue function, which is the crucial

part of the algorithm. Separation of the convolved signals in the complex cepstrum is complicated in the presence of low signal to noise ratios as in the fMRI cases [37].

Blind deconvolution is applied mainly to image restoration. Image restoration is the process of recovering the original image from the degraded image and also understanding the image without any artifacts errors. The main idea is to deconvolve the blurred image with the blur kernel function that represents the distortion during the imaging process. An original image is degraded and blurred using degradation model to produce the blurred image for this deconvolution application. The blurred image is given as an input to the deblurring algorithm, which is blind deconvolution in this approach. An iterative blind deconvolution algorithm is applied in which both the original image and the blurring function is estimated initially. The restoration quality of the algorithm is visually and quantitatively compared with other algorithms. [38]. In this work only the Gaussian noise is used to blur the original image. In fMRI applications, there is obviously more than one type of noise integrated to the degraded image, which is much more difficult to deal with.

As an image processing application, blind deconvolution is used to enhance scanning electron microscope images after magnification. A self-deconvolving data reconstruction algorithm is applied which is a non-iterative method that uses a Wiener filter and a special smoothing function. The methodology yields reasonable results as long as the images have high resolution [39]. That means high SNR is needed for the success of the algorithm, so fMRI signals would not yield reasonable results with this application.

Blind deconvolution process has also been applied to audio signals. An acoustic signal is produced by a conventional computer speaker, and simultaneously recorded by the computer microphone. This action imposes an unknown frequency response upon the signal. A transfer function is derived from the two signals and used to restore the degraded signal. The process compares the magnitude of the data in Fourier space to the

same quality of a specified reference data set. A filter function is derived from the comparison and used as a transfer function for restoring the original data. The qualitative experiments verified that the algorithm can be applied to acoustic waveforms to restore the frequency characteristics of the signal. The success of the restoration is dependent on the choice of an appropriate reference signal [40].

Another approach of blind deconvolution is bar code reconstruction. The problem is to recover a bar code from the noisy signal detected by a bar code reader. This process is modeled as the convolution of the pure bar code signal with a Gaussian kernel of unknown amplitude and standard deviation. The algorithm minimizes energy functional via gradient descent iteratively. As it is an iterative method, initial estimations are critical for the results. In test experiments, the original bar code signal was corrupted by convolving it with a kernel of known standard deviation, followed by the addition of some noise and performance is evaluated based not on the speed of reconstructions but their accuracies [41].

Blind deconvolution has also been applied for eliminating the echo affect in a speech signal propagating from a speaker to several distant microphones within a reverberant enclosure. The method aims to improve the performance of an automatic speech recognition system. The proposed algorithm assumes that a source signal is measured by several sensors after propagating through channels and being corrupted by additive noise. The channels are modeled as finite impulse response linear filters. The applied blind deconvolution algorithm consists of two steps. First, the propagation channels are blindly estimated from the sensor signals using maximum likelihood (ML) approach. Next, estimated channel filters are used in exact deconvolution to obtain to the pure speech signals. Although a fast implementation has been derived by this approach, the computational cost is prohibitive for large order channel filters [42].

Detection of failure in rotating machinery is investigated using blind deconvolution. The observed signal from the bearing condition monitor is often corrupted by noise during the transmission process. The expected time intervals between the impacts of faulty bearing components signals are analyzed using the blind deconvolution technique to recover the source signal. Blind deconvolution refers to the process of learning the inverse of an unknown channel and applying it to the observed signal to recover the source signal of a damaged bearing. The estimation time period between the impacts is improved by using the technique and consequently provides an approach to identify a damaged bearing. The procedure to obtain the optimum inverse equalizer filter is addressed to provide the filter parameters for the blind deconvolution process. It is shown that the blind deconvolution behaves as a notch filter to remove the noise components. Hence, blind deconvolution is said to be used as a signal enhancing technique for lifetime testing of a rolling element bearing until failure occurs. Symptoms of failure is known by the experimenter in advance and enhancing the observed signal by blind deconvolution makes the detection of these symptoms possible and as early as possible [43].

In seismic deconvolution, the geological layers of earth are modeled with different acoustic impedance. The sequence of reflection coefficients is estimated corresponding to the various types of layer models. The received signal is made up of echoes produced at different layers of the model in the response to the excitation, which is in the form of short-duration pulse. The extraction of the excitation waveform associated with the received signal is usually unknown and blind deconvolution is applied to the observed or measured signal to recover the source of excitation. The impulse response of the layered earth model is viewed as equally spaced time sequence of reflection coefficients [44], [45].

Blind deconvolution has also been concerned with video sequences. The method processes the video by applying temporal windows, masking out regions of

misregistration, and minimizing a regularized energy function with respect to the high-resolution frame and blurs, where regularization is carried out in both the image and blur domains. The proposed algorithm performs resolution enhancement and deblurring of video sequences [46].

Besides, blind deconvolution is used to measure travel time from two sets of spatially separated loop detectors to provide vital information for traffic monitoring, management and planning. Blind deconvolution is applied to reverse the effect of convolution by the loop detector and to separate the original vehicle signatures from the loop output. Since neither the impulse response of the loop detector nor the original vehicle signature is known, an iterative blind deconvolution technique is developed to obtain the original vehicle signature but the performance of the algorithm depends heavily upon the initial estimations [47].

The examples of blind deconvolution applications can be increased but the main objective behind all blind deconvolution applications is the same. The objective of blind deconvolution is to reconstruct the original signal from a degraded observed signal without the knowledge of either the original signal or the degradation process. Therefore blind deconvolution well-fits to our fMRI problem. Because the fMRI signal is degraded and it contains many different blurring components due to physiological factors, fluctuations in the blood flow, blood volume and oxygen metabolism; changes in behavioral performance and cognitive strategy; and variability in neural activity based on non-task related brain processes. These blurring signals have to be removed from fMRI so that the original signal, HRF, representing neural activity can be extracted. Furthermore, blind deconvolution is a nonlinear process which makes it suitable for HRF, which has nonlinear characteristics. It is known that when two stimuli are presented successively, the resulting hemodynamic response is less than the sum of responses to the two stimuli separately [14]. So, linear approaches are inadequate for many fMRI analyses. Also in blind deconvolution no priori assumptions are required for

HRF. Moreover, usually in fMRI data analysis, it is common to assume that the instant a stimulus is administered, it is perceived by the subject. However, there are always attentional processes which intervene with stimulus delivery. When we use complex or natural stimuli, then sometimes we cannot define the stimulus administration time in a straightforward manner. By means of a properly developed blind deconvolution algorithm HRF can be obtained without the knowledge of the stimuli function. For these advantages, blind deconvolution became the motivation for us while extracting the HRF.

After obtaining HRF by means of blind deconvolution, we will be able to identify whether a given specific voxel exhibits significant task-related signal changes. This approach is conducted independently on all voxels. Nevertheless, when functional relations between different voxels are considered, blind deconvolution would be cumbersome, since it is specialized for determining the underlying neural activity of one specific voxel. For classifying the voxels according to their functional similarities, many classification and clustering techniques are developed so that activated voxels can be separated from non-activated voxels. But all classification techniques require a label for each data in the training set. In fMRI analysis it would be very difficult to use previously collected fMRI data sets as a training set since the fMRI signal depends on many factors such as experimental design, subject, and region of the brain activated. For these reasons we used clustering to classify voxels. In the following section clustering will be discussed.

2.2 Approaches to Clustering of fMRI

Although fMRI represents a powerful technique for visualizing rapid and fine activation patterns of the human brain, the low signal-to-noise ratio (SNR) and confounding sources of artifacts in the fMRI time series, make the detection of brain activation a challenging task. After extracting HRF from fMRI signal, the SNR increases since the noise in the observed fMRI data is filtered via blind deconvolution. So, detection of brain activation and classifying the voxels according to their functional similarities become more feasible. We used clustering for activation detection with the aim of separating time series into several patterns according to their similarities.

Clustering methods group data samples based only on information found in the data that describes the objects and their relationships. The goal is to obtain data within a group be similar to one another while being different from the data in other groups. The greater the similarity or homogeneity within a group and the greater the difference between groups, the better or more distinct is the clustering.

In fMRI data clustering, mostly hierarchical, k-means and spectral clustering methods are used.

2.2.1 Hierarchical Clustering

In hierarchical clustering, each sample is initially considered a member of its own cluster, after which clusters are recursively combined in pairs according to some predetermined condition until eventually every point belongs to a single cluster. Figure 2.2 shows an illustration of hierarchical clustering. One problem in hierarchical clustering is how to choose which clusters or partitions to extract from the hierarchy

since display of the full hierarchy is not really appropriate for large data sets. Also it suffers from computational cost especially when the data set is too large.

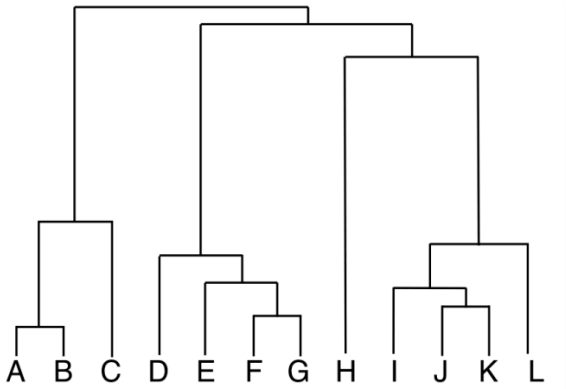


Figure 2.2 Hierarchical Clustering

In [20] a hierarchical clustering algorithm is applied to resting-state fMRI in which no explicit tasks are performed. Temporal coherence of low frequency oscillations in the brain regions characterizes the functional connectivity. The aim is to find clusters whose voxel members have high cross correlation coefficients that indicate low frequency synchronous fluctuations in the fMRI signal. Synchrony infers to functional connectivity. The method does not require prior knowledge of cluster centers or the number of clusters present in the data. The algorithm is based on connecting data by a single link. As a similarity measure to group voxels into clusters for functional connectivity, a cross correlation similarity measure is used. All clusters are obtained by considering only frequency contributions. However, analysis of possible motion contribution for these clusters gave insignificant numbers and does not provide a sufficient explanation. Also, single linkage hierarchical clustering tends to produce long-chained clusters. That means it assumes that data conform a certain shape. However, in our problem the fMRI data set does not have a certain shape. We analyze task-related fMRI data which has more complicated structure than resting-state fMRI data since there are signals due to the task related neural activity in addition to the resting state activities in the brain, so our data set scatters amorphous. Besides, the data sets in our

experiments are too large for hierarchical clustering while resting state fMRI is usually driven from a small seed region of interest. Because of these, in our approach hierarchical clustering is not applicable.

2.2.2 K-means Clustering

In k-means clustering, every data sample is initially assigned to a cluster in a random way. Samples are then iteratively transferred from cluster to cluster until some criterion function is minimized. Once the process is complete, the samples will have been partitioned into separate compact clusters.

K-means procedure follows a simple and easy way to classify a given data set through a certain number of clusters, *k*, fixed a priori. The main idea is to define *k* centers, one for each cluster. These centers initially should be placed carefully because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. Then, each data point is assigned to the nearest center. After placing all the data to a cluster according to their closeness to the center, the early clustering is done. After that, new cluster centers are determined and the data points are again assigned to the cluster that has the closest center. As a result of these iterations, the k centers change their location step by step until no more changes are done. In other words centers do not move any more.

This algorithm aims at minimizing an objective function, in this case a squared error function;

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 \tag{2.16}$$

where $\left\| x_i^{(j)} - c_j \right\|$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster center c_j , $\|.\|$ is an indicator of the distance of the n data points from their respective cluster centers.

With a large number of variables, if *k* is small, k-means may be computationally faster than hierarchical clustering. K-Means may produce tighter clusters than hierarchical clustering, especially if the clusters are spherical. Distance between cluster centers is the furthest when data scatter spherically, so center points can be determined easily. But if the data set does not have a spherical shape such as it is elliptical, the data may be closer to a different cluster center than it belongs to. So, we may put data to wrong clusters.

Although it can be proved that the center determination procedure will always terminate, the k-means algorithm does not necessarily find the most optimal configuration, corresponding to the global objective function minimum. The algorithm is also significantly sensitive to the initial randomly selected cluster centers. The k-means algorithm can be run multiple times to reduce this effect. Different initial *k* values affect the performance of the clustering, so it is difficult to compare the quality of the clusters produced. Also, fixed number of clusters can make it difficult to predict what *k* should be.

The obtained clusters after k-means clustering are not stable enough in repeated runs with the same data sets. A clustering result can be characterized by the quantization errors, i.e., the distances between representants and data points, and the reduction degree, i.e., the number of data points divided by the number of cluster centers k. Since the tradeoff between the quantization errors and the reduction degree varies from data set to data set, the *k* value in k-means analysis must often be optimized iteratively by a result driven search for each data set [48].

K-means is applied to a large data set of fMRI to divide the data as extrinsic and intrinsic. Extrinsic areas are associated with the external applied stimuli, while intrinsic areas are activated internally not due to the external stimuli. However, some data do not fall into either the extrinsic or intrinsic subdivisions since the functional characteristics of the intrinsic system cannot fully established [18].

Also the k-means algorithm intrinsically assumes that the data are scattered with respect to a spherical distribution, i.e, for example normal distribution with identity covariance. As the distribution of data gets further from spherical shape, the k-means clustering algorithm performs poorer. Figure 2.3 illustrates an optimum data distribution for k-means algorithm.

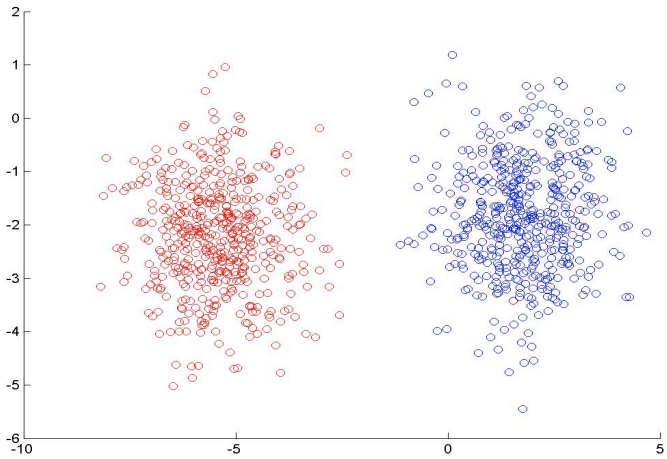


Figure 2.3. K-means clustering with spherical distributed data

As it is shown in Figure 2.3, k-means perfectly identify the two clusters. On the other hand, if the two clusters of data are scattered as follows, ‘banana’ shape, then k-means fails since it does not deal with the geometry. Although two clusters in Figure 2.4 are far from having Gaussian distribution, k-means still tries to find two centers and assign labels to points by checking to which center they are closer.

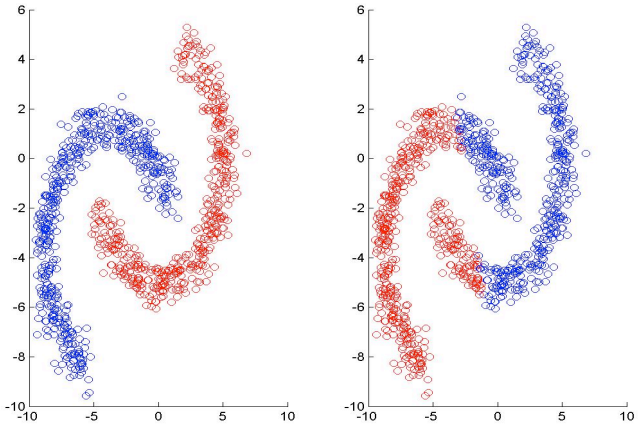


Figure 2.4 Failure of k-means with data in different manifolds

In order to cope with this issue, one has to first consider the geometry of the data by probably defining an appropriate metric between points, then, if possible, transform it to a space wherein standard clustering algorithms can perform well. For instance, there is another clustering algorithm named 'spectral clustering' which first spectrally transform the data by constructing a neighborhood graph and then applies k-means. It is widely used in clustering applications, which has been reported repeatedly to give good clustering results. Even after the transformation one might need to use a more sophisticated clustering than k-means for which Expectation Maximization (EM) clustering can be proposed. It is the generalization of k-means and has the capability of dealing with elliptic distributions as well, not only spherical. In the following we first explain the details of EM and how it works, then Spectral Clustering. In this thesis for fMRI clustering, we use for aforementioned reasons Spectral Clustering with EM.

2.2.3 Expectation Maximization (EM)

Expectation Maximization (EM) is a generalized version of k-means clustering. EM finds clusters by determining a mixture of Gaussians that fit a given data set. Each Gaussian has an associated mean and covariance matrix. The prior probability for each Gaussian is the fraction of points in the cluster defined by that Gaussian. These parameters can be initialized by randomly selecting means of the Gaussians, or by using the output of k-means for initial centers. The algorithm converges on an optimal solution by iteratively updating values for means and variances of the clusters.

Let $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ be a data set of n independent observed signal from a mixture of two multivariate normal distributions of dimension *d*, and let $\mathbf{z}=(z_1, z_2, \ldots, zn)$ be the variables that determine the component to which the observed signals belong.

$$\mathbf{X_i}|(\mathbf{Z_i}=1) \sim N_d(\mu_1, \Sigma_1) \text{ and } \mathbf{X_i}(\mathbf{Z_i}=2) \sim N_d(\mu_2, \Sigma_2) \tag{2.17}$$

where

$$P(\mathbf{Z_i}=1) = \tau_1 \text{ and } P(\mathbf{Z_i}=2) = \tau_2 = 1-\tau_1 \tag{2.18}$$

We try to estimate the unknown parameters of the Gaussian clusters of which;

$$\boldsymbol{\theta} = (\tau, \mu_1, \mu_2, \Sigma_1, \Sigma_2) \tag{2.19}$$

Also the likelihood function can be given as:

$$L(\boldsymbol{\theta};\mathbf{x},\mathbf{z}) = P(\mathbf{x},\mathbf{z}|\boldsymbol{\theta}) = \prod_{i=1}^{n}\sum_{j=1}^{2}\mathbf{I}(z_i = j)\tau_j f(x_i;\mu_j,\Sigma_j) \tag{2.20}$$

where **I** is the indicator function which is defined as:

$$\mathbf{I}(x = i) = \begin{cases} 1 & if\ \ x = i \\ 0 & if\ \ x \neq i \end{cases} \tag{2.21}$$

and *f* is the probability density function of a multivariate normal. In the exponential form this likelihood function can be written as;

$$\begin{aligned} L(\boldsymbol{\theta};\mathbf{x},\mathbf{z}) &= P(\mathbf{x},\mathbf{z}|\boldsymbol{\theta}) \\ &= \exp\left\{\prod_{i=1}^{n}\sum_{j=1}^{2}\mathbf{I}(z_i = j)\left[\begin{array}{l}\log\tau_j - \frac{1}{2}\log|\Sigma_j| \\ -\frac{1}{2}(x_i-\mu_j)^T\Sigma_j^{-1}(x_i-\mu_j) - \frac{d}{2}\log(2\pi)\end{array}\right]\right\} \end{aligned} \tag{2.22}$$

E-Step

The parameters of the Gaussians are initially estimated in this step. The estimation of the parameters in the *t* th iteration, $\boldsymbol{\theta}^{(t)}$, are used to calculate the conditional distribution of the *Zi*. It is determined by Bayes' theorem weighted by $\boldsymbol{\tau}$.

$$\mathbf{T}_{j,i}^{(t)} = P(\mathbf{Z}_\mathbf{i} = j|\mathbf{X}_\mathbf{i} = x_i;\boldsymbol{\theta}^{(t)}) = \frac{\tau_j^{(t)} f(x_i;\mu_J^{(t)},\Sigma_J^{(t)})}{\tau_1^{(t)} f(x_i;\mu_1^{(t)},\Sigma_1^{(t)}) + \tau_2^{(t)} f(x_i;\mu_2^{(t)},\Sigma_2^{(t)})} \tag{2.23}$$

Then the expectation of $\boldsymbol{\theta}$ is given by:

$$Q\left(\theta\middle|\theta^{(t)}\right) = E\left[\log L(\boldsymbol{\theta};\mathbf{x},\mathbf{Z})\right]$$
$$= \sum_{i=1}^{n}\sum_{j=1}^{2}\mathbf{T}_{j,i}^{(t)}\left[\log\tau_j - \frac{1}{2}\log\left|\Sigma_j\right| - \frac{1}{2}\left(x_i-\mu_j\right)^T\Sigma_j^{-1}\left(x_i-\mu_j\right) - \frac{d}{2}\log(2\pi)\right] \quad (2.24)$$

M-Step

In the Maximization (M) step the aim is to find the parameters that maximize the expected log-likelihood found in the E step. Since $Q\left(\theta\middle|\theta^{(t)}\right)$ is quadratic, taking the derivative and find where it equals to zero gives us the maximizing values of θ. $\boldsymbol{\theta},\boldsymbol{\tau},(\mu_1,\Sigma_1)\ and\ (\mu_2,\Sigma_2)$ can be maximized independently from each other since they all appear in separate linear terms.

Firstly we maximize τ, which has the constraint of $\tau_1+\tau_2=1$.

$$\tau^{(t+1)} = \arg\max Q\left(\theta\middle|\theta^{(t)}\right)$$
$$= \arg\max_{\tau}\left\{\left[\sum_{i=1}^{n}\mathbf{T}_{1,i}^{(t)}\right]\log\tau_1 + \left[\sum_{i=1}^{n}\mathbf{T}_{2,i}^{(t)}\right]\log\tau_2\right\} \quad (2.25)$$

This has binomial distribution form and can be restated as:

$$\tau_j^{(t+1)} = \frac{\sum_{i=1}^{n}T_{1,i}^{(t)}}{\sum_{i=1}^{n}T_{1,i}^{(t)}\left(T_{1,i}^{(t)}+T_{2,i}^{(t)}\right)} = \frac{1}{n}\sum_{i=1}^{n}T_{j,i}^{(t)} \quad (2.26)$$

Next estimates of (μ_1, Σ_1) can be similarly calculated to maximize the log-likelihood function:

$$\begin{aligned}\left(\mu_1^{(t+1)}, \Sigma_1^{(t+1)}\right) &= \arg\max_{\mu_1,\Sigma_1} Q\left(\boldsymbol{\theta}\middle|\boldsymbol{\theta}^{(t)}\right) \\ &= \arg\max \sum_{i=1}^{n} \mathbf{T}_{1,i}^{(t)} \left\{ -\frac{1}{2}\log|\Sigma_1| - \frac{1}{2}\left(x_i - \mu_1\right)^T \Sigma_1^{-1}\left(x_i - \mu_1\right) \right\}\end{aligned} \tag{2.27}$$

This has the normal distribution form, so:

$$\mu_1^{(t+1)} = \frac{\sum_{i=1}^{n} T_{1,i}^{(t)} x_i}{\sum_{i=1}^{n} T_{1,i}^{(t)}} \text{ and } \Sigma_1^{(t+1)} = \frac{\sum_{i=1}^{n} T_{1,i}^{(t)} \left(x_i - \mu_1^{(t+1)}\right)\left(x_i - \mu_1^{(t+1)}\right)^T}{\sum_{i=1}^{n} T_{1,i}^{(t)}} \tag{2.28}$$

and by symmetry:

$$\mu_2^{(t+1)} = \frac{\sum_{i=1}^{n} T_{2,i}^{(t)} x_i}{\sum_{i=1}^{n} T_{2,i}^{(t)}} \text{ and } \Sigma_2^{(t+1)} = \frac{\sum_{i=1}^{n} T_{2,i}^{(t)} \left(x_i - \mu_2^{(t+1)}\right)\left(x_i - \mu_2^{(t+1)}\right)^T}{\sum_{i=1}^{n} T_{2,i}^{(t)}} \tag{2.29}$$

These estimated parameters are used in the next E step to determine the distribution of the Gaussian variables Z. These iterations continue until the estimated parameters converge to stable values and so the change in their values becomes negligible.

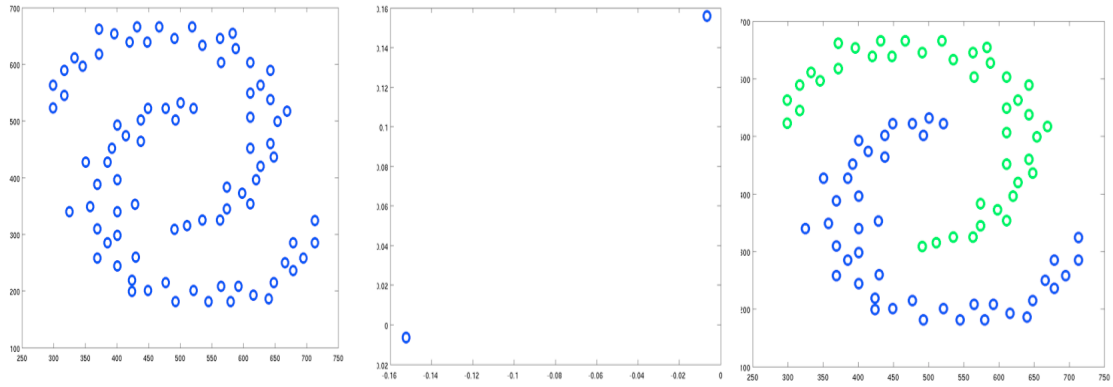
2.2.4 Spectral Clustering

In recent years, spectral clustering has become one of the most popular modern clustering algorithms. It is widely used in machine learning, exploratory data analysis, computer vision, speech processing and statistics [49], [50]. It is simple to implement, can be solved efficiently by standard linear algebra software, and very often outperforms traditional clustering algorithms such as the k-means algorithm, EM clustering, PCA. The reason behind is that these traditional techniques do not consider the intrinsic geometry of the data.

On the other hand, Laplacian Eigenmaps, which is the essence of Spectral Clustering, builds a similarity graph from neighborhood information of the data set. Each data point serves as a node on the graph and connectivity between nodes is governed by the proximity of neighboring points (using e.g. the k-nearest neighbor algorithm) along with the weight of the edges, if constructed based on whether they are declared to be neighbors, between them. This is in accordance with the intuition behind the clustering which is that one wants to divide the data points into several groups such that points in the same group are similar and points in different groups are dissimilar to each other. The problem of clustering can then be reformulated using the similarity graph: we want to find a partition of the graph such that the edges between different groups have very low weights (which means that points in different clusters are dissimilar from each other) and the edges within a group have high weights (which means that points within the same cluster are similar to each other). The graph, thus, generated can be considered as a discrete approximation of the low dimensional manifold in the high dimensional space. Minimization of a cost function based on the graph ensures that points close to each other on the manifold are mapped close to each other in the low dimensional space, preserving local distances wherein the traditional methods, then, are expected to perform well. In short, Spectral Clustering via the Laplacian Eigen Maps by connecting the data points only to its neighbors, it estimates the intrinsic geometry of the data. As a result, it

has been repeatedly reported in the literature that it outperforms traditional clustering algorithms, which usually assume Gaussianity in the data and so misses possibly the geometric structures.

When there are no exact Gaussian clusters, that means the data do not scatter spherically in a cluster and most of the data in that cluster do not place close to the cluster center, the data can be grouped via spectral clustering by transforming the data to a proper feature space. The idea behind the spectral clustering is to replace a group of similar variables by a cluster center, which becomes a new derived feature. The new derived feature then is treated as a representative for the whole cluster as the new input for the classifier. The main trick is to change the representation of the abstract data points, so that clusters can be trivially detected in the new representation. In particular, the simple k-means clustering algorithm has no difficulties to detect the clusters in this new representation. The spectral clustering will greatly reduce the number of features and meanwhile without losing much information. In Figure 2.5a the data scattered in a spiral shape. There are two “banana-shaped” structures one inside the other. Spectral clustering identifies these two different behaviors and selects these behaviors as feature representatives of each banana-shape in Figure 2.5b. Then by means of a simple clustering algorithm such as k-means, these two data can be separated easily since they are far enough from each other in the transformed space. Finally as seen in Figure 2.5c the data set is clustered according to their feature similarities.



a. Original Data b. Transformed Data c. Spectral Clustering

Figure 2.5 Spectral Clustering

Figure 2.5 shows a famous data set, which is known as the 'banana data set' in the machine learning literature. It has a special geometry of banana shape wherein an assumption of Gaussianity badly fails. And precisely for this reason other algorithms like k-means or EM clustering perform poorly in terms of the clustering of these two banana shape data sets. However, once the geometry is exploited through the Laplacian Eigenmaps in Spectral Clustering as in Figure 2.5b, and the data points are mapped to a low dimensional space, therein, then by means of a simple clustering algorithm like k-means, these two data can be separated easily since they are far enough from each other in the transformed space.

In the spectral clustering application, given a set of data points $x_1,...x_n$ firstly the similarities s_{ij} between all pairs of data points x_i and x_j are calculated by using a proper distance metric. After similarities between the data points are known which have already been calculated using a proper metric, the data is presented in form of what is known as the similarity graph G = (V, E). The data points x_i are represented by the vertices v_i in this similarity graph. Two vertices are connected if the similarity s_{ij} between the corresponding data points x_i and x_j is larger than a determined threshold value, and the connection edge is weighted according to s_{ij}.

Similarity graphs are essential for modeling the local neighborhood relationships between the data points and can be developed in different forms:

The ε-neighborhood graph: In these type graphs, the vertices v_i and v_j are connected if the related distance is smaller than ε, otherwise they are not connected.

k-nearest neighbor graphs: In these type graphs, the vertex v_i is connected with v_j if v_j is among the k nearest neighbors of v_i. Since the neighborhood relationship may not be symmetric, these graphs lead to a directed graph. This graph can be made undirected by connecting vi and vj with an undirected edge if vi is among the k-nearest neighbors of vj or if vj is among the k-nearest neighbors of vi. The second method to make the graph undirected is to connect vertices vi and vj if both vi is among the k-nearest neighbors of vj and vj is among the k-nearest neighbors of vi. After connecting the appropriate vertices, the edges are weighted by the similarity of the adjacent points.

The fully connected graph: In that type of graph all point are connected with each other, with all the edges are weighted according to their similarity.

Using the proper similarity graph, partitions of the graph can be created such that the edges between different groups have a very low weight to indicate the dissimilarity of clusters and the edges within a group have high weight meaning that the points in the same cluster are similar.

Let G= (V, E) be a similarity graph with vertex set V= $\{v_1,\ldots,v_n\}$. The graph G can be weighted, that means each edge between two vertices v_i and v_j carries a nonnegative weight $w_{ij}\geq 0$. Using these weights a weight matrix is defined as $W= (w_{ij})_{i,j=1,\ldots,n}$. If vertices v_i and v_j are not connected then $w_{ij}=0$. G is also undirected which yields $w_{ij}= w_{ji}$. The degree of a vertex $v_i \in V$ is defined as

$$d_i = \sum_{j=1}^{n} w_{ij} \tag{2.30}$$

The degree matrix D is then defined as the diagonal matrix with the degrees $d_1, \ldots, d_n$ on the diagonal.

Firstly, the graph Laplacian matrix **L** is calculated using weight matrix W that has already constructed by means of chosen similarity metric. That is;

$$\mathbf{L} = \mathbf{D} - \mathbf{W} \tag{2.31}$$

The matrix L satisfies the properties below:

- For every vector $f \in \mathbf{R}^n$, it is satisfied that

$$f' Lf = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2 \tag{2.32}$$

- **L** is symmetric, semi-definite and positive.
- The smallest eigenvalue of **L** is 0; the corresponding eigenvector is the constant one vector.
- **L** has n non-negative, real-valued eigenvalues $0=\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$

After calculating the Laplacian matrix **L**, the generalized eigenvalue problem should be solved which is;

$$\mathbf{L}v = \lambda \mathbf{D}v \tag{2.33}$$

This eigenvalue problem yields eigenvectors of $v_1, \ldots, v_n$ as solution.

If there exists k clusters in the data it would be better to take first k eigenvectors $v_1,\ldots v_k$ as columns to form a **V** matrix in the form of $\mathbf{V} \in R^{nxk}$. In the generated V matrix, i^{th} row vector y_i, will represent the corresponding data x_i. Then these points $(y_i)_{i=1,\ldots,n}$ can be clustered into k clusters by means of a proper clustering algorithm, like k-means, fuzzy c-means, expectation maximization, etc.

To sum up, the goal of spectral clustering is to separate a finite unlabeled data set into a finite and discrete labeled set of "natural", hidden data structures. Clustering is an ill-posed problem from the mathematical point of view, because what is "natural" is usually decided by human judgment since no labeled data are available, but despite of this, clustering algorithms are frequently applied to gain insight to the structure of the data.

Spectral clustering is used as a machine learning technique in medical arena among which investigating relations between a gene graphs constructed using microarray experiments and the one based on the text corpus is envisaged. The performance of a spectral clustering method using text data only, using microarray data only and using the technique for integrating both knowledge bases is discussed [51]. The learning technique addresses the unsupervised learning problem of finding meaningful clusters co-occurring in both knowledge bases.

Spectral clustering is examined for breaking up the behavior of a multi-agent system in space and time into smaller, independent elements. The clustering technique is extended into temporal domain [51]. For example, in a football match grouping the agents into two teams can be discussed in this research. There are many indicators of interaction to separate these agents. The agents participated in performing to score a goal can be determined to put in the same cluster. It means we need to identify temporal and spatial boundaries between these elements in a very noisy environment. Since only agent positions is not enough for catching the behavioral aspects of the multi-agent system, knowledge about events of multi-agent interaction with different importance is exposed

by developed spectral clustering technique, so events are assumed to have relative importance measure. Besides, spectral clustering is applied to a strategic computer game to detect the events covering some important interaction.

Spectral clustering is also used in genomics for clustering homologous proteins when only their sequence information is known. Spectral clustering, as a global method, has been compared with the local clustering methods of which have some limitations. The local methods are said to be based on simply thresholding a measure related to the distance between the sequences. They assign a protein to a cluster taking into account only the distances between that protein and the other proteins in the set. Instead, as a global method, spectral clustering assigns a protein to a cluster taking into account all the distances between every pair of proteins in the set. In the performed experiments, it is concluded as the quality of the clusters as quantified by a measure that combines sensitivity and specificity was consistently better when spectral clustering is used [53].

Especially in biological data analysis the large data sets are difficult to handle due to time and memory limitations. Spectral clustering is modified by adding an information preserving sampling procedure and applying a post-processing stage. Flow cytometry data as an example of large, multidimensional data containing potentially hundreds of thousands of data points is used as test data. Modified method reduces the size of input for spectral clustering algorithms and consequently they can now be efficiently applied on flow cytometry data in spite of its large size. In conclusion, the spectral clustering approach demonstrates significant advantages in proper identification of populations with non-elliptical shapes, low-density populations close to dense ones, minor subpopulations of a major population, rare populations, and overlapping populations. Moreover, applying the modified spectral clustering method to other biological data such as microarrays and protein databases may result in significant improvements in gene expression and protein classification [54].

Spectral clustering algorithms also have found application in tissue classification in MRI. An empirical evaluation of a stochastic sampling approach to modeling voxel-to-voxel relationships for spectral clustering is exposed [55]. Stochastic sampling captures sufficient intensity structure to give plausible tissue classification in 3D brain MRI. Magnetic Resonance images are classified into three common tissue types which are Grey- Matter (GM), White-Matter (WM), Cerebro-Spinal Fluid (CSF) and also the background in the field of view.

Spectral clustering is also applied to fMRI time series [56]. It is compared with other clustering methods by using a modified version of a common fMRI clustering metric obtained by the cross-correlation of the fMRI signal with the experimental protocol signal [56]. Experiments are performed with synthetic and real fMRI data. Real fMRI data are picked from 300 voxels that correspond to about 5% of the brain voxels. The modified clustering metric gives satisfactory results though, but spectral and stochastic clustering methods had better to be evaluated further in the presence of more inactive noise voxels.

Spectral clustering has the advantage of performing well with non-Gaussian clusters as well as being easily implementable. It is also non-iterative with no local minima. Besides it is not restricted to convex regions of similarity and the robustness to noise makes spectral clustering attractive. Hence in this thesis, spectral clustering is chosen to group the fMRI data according to their activation states (i.e. active, passive). Furthermore another reason for preferring spectral clustering is that besides the used distance measure, it requires no assumptions about spatial location and extension of activation sites or the shape of the expected activation time series. Spectral clustering often outperforms other clustering techniques, since it captures the natural structure of the data set. This implies that spectral clustering can identify clusters with complex signal geometries. Moreover, spectral clustering can be implemented efficiently even for

large data sets. As all the clustering algorithms, spectral clustering can only be as good as the distance measure used.

2.2.4.1 Distance Measure

The success of any clustering algorithm depends heavily on the choice of the distance measure, which will give the similarity of two elements in the data set. This will influence the shape of the clusters, as some elements may be close to one another according to one distance but farther away according to another. The problem at hand is to meaningfully quantify the similarity between two fMRI time series.

There are many distance functions, the most known being the Euclidean distance.

Euclidean Distance

Euclidean distance is the distance between two points defined as the square root of the sum of the squares of the differences between the corresponding coordinates of the points. It is the function $d : R^n x R^n \rightarrow R$ that assigns to any two vectors in Euclidean n-space $x = (x_1 \ldots x_n)$ and $y = (y_1 \ldots y_n)$ the number;

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + \ldots + (x_n - y_n)^2} \tag{2.34}$$

and so gives the Euclidean distance between any two vectors in R^n.

Euclidean distance can distinguish data obtained from similar task paradigms, having large and discernible timing variability, of the order of a few seconds. However, if the timing difference is small, of the order of a few tens of milliseconds, Euclidean distance is not successful to identify the similarities of the data. Moreover, Euclidean distance

does not accurately delineate the interference of noise points in fMRI signals [21]. So, in the literature there are not many applications that use the conventional Euclidean distance. It does not define a similarity measure of the data, but instead it measures the direct distance between the data. So, in this thesis Hausdorff distance is used for similarity measure.

Hausdorff Distance

Hausdorff distance is a metric used especially for object matching and shape detection which measures the extent to which each point of a model set lies near some point of an image set and vice versa. It attracts many engineers, scientists since it is insensitive to affine invariants like translation, rotation and scaling [57].

In clustering analysis, Hausdorff distance is used as distance measure in this thesis. Hausdorff distance computes the distance between two data sets. Given two finite point sets A=$\{a_1,\ldots, a_n\}$ and B=$\{b_1,\ldots, b_n\}$ the distance from any point $a \in A$ to the set B is calculated with an underlying norm $\|.\|$ (e,g., the L_2 norm or Euclidean norm) which can be defined as;

$$d(a,B) = \min_{b \in B} d(a,b) = \min_{b \in B} \|a-b\| \tag{2.35}$$

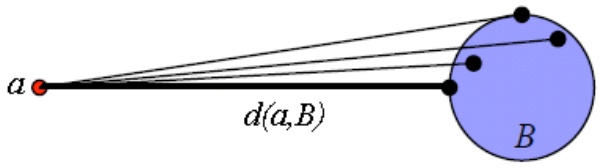


Figure 2.6 Distance from a point to a set.

Then, the distance from A to B which is called as directed Hausdorff distance, h(A,B) is obtained using;

$$h(A,B) = \max_{a \in A} d(a,B) = \max_{a \in A} \min_{b \in B} d(a,b) \tag{2.36}$$

Basically h(A,B) determines the point $a \in A$ which is the farthest to any point of B, to any point of B and computes the distance from a to the point $b \in B$ which is the nearest point to a. That means it searches each point of A to find out its distance to the nearest point of B, and then selects the largest distance of those. The point of A yielding the largest distance is the most mismatched point of A.

Then, the Hausdorff distance is the maximum of $h(A,B)$ and $h(B,A)$. That can be represented as;

$$H(A,B) = \max\left(h(A,B), h(B,A)\right) \tag{2.37}$$

Hausdorff distance can be thought to measure the degree of mismatch between two sets by measuring the distance of the point of A that is farthest from any point of B and vice versa.

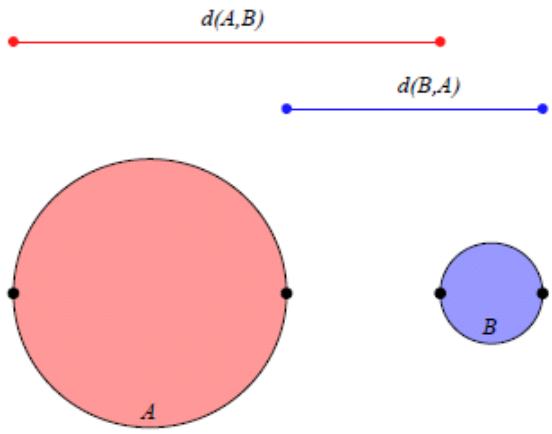


Figure 2.7 Distance from a set to other set

Equivalently, Hausdorff distance can be reposed by means of the dilations by the balls of the space, which gives a more intuitive understanding. If $B_\varepsilon(x) = \{y : \|y - x\| \leq \varepsilon\}$ is the

compact ball centered at a point *x*, then one-way Hausdorff distance from a set A to a set B is given by:

$$h(A,B) = \arg\min\left\{\varepsilon : U_{a_i \in A} B_{\in}(a_i) \supseteq B\right\} \tag{2.38}$$

So the Hausdorff distance basically is the minimum amount of dilation of the set A by balls of the space to cover the set B, and vice versa. This directly reveals an important disadvantage of using Hausdorff distance which is that when there exists an outlier in set A or set B, in other words when – for instance- there exists a separated point, this can make the Hausdorff distance unnecessarily large and in general very sensitive to outliers. In order to overcome this, one-way Hausdorff distance can be modified to cover the target set to a significant fraction but not completely:

$$h_a(A,B) = \arg\min\{\varepsilon : U_{a_i \in A} \left| B_{\in}(a_i) \cap B \right| \supseteq a|B|\} = \max^{a}{}_{a \in A}\, d(a,B)\} \tag{2.39}$$

where the parameter α controls what fraction of the target set is desired to be covered and it is, usually, close 1. This is equivalent to taking not the very maximum but the $(1-a)*|B|$ 'th maximum in (2.39).

In our application we will use this modified version of Hausdorff distance in which basically we do not take the maximum distance but instead we take α^{th} maximum to prevent the misleading effects of the large distances due to the outliers (instant noisy peaks).

Hausdorff distance is used for human face recognition by obtaining the edge map of a facial image, which contains crucial information about its shape and structure. Hausdorff distance is modified to determine the weight function derived from the spatial information of the human face [58]. Hausdorff distance is also used in hybrid image matching with a similarity measure, which combines a modified Hausdorff distance with

the normalized image gradient matching. This integrated image similarity measure that uses Hausdorff distance is applied to the problem of face recognition under different illumination conditions [59].

A method for determining affine transformations that bring a model into close correspondence with a portion of an image is applied via modified version of the Hausdorff distance [60]. Even in the presence of occlusion, spurious points, positional uncertainty, strong restrictions on the transformation range and the presence of multiple objects, Hausdorff distance model detects the true object and determines the location of the object accurately. However, the model runs quite slowly as the image set becomes larger.

The shape-matching problem with and without constraints on the allowed transformations under the Hausdorff distance is also studied in [61]. Computation of exactly or approximately the smallest Hausdorff distance over all possible rigid motions in two and three dimensions is examined. In many biological applications especially in molecular biology, shapes can be approximated by a finite union of balls. For example, proteins with similar shapes are likely to have similar functionalities; therefore classifying proteins based on their shapes is an important problem in computational biology. So, the main type of input it the application is the ball shapes

Hausdorff Distance is applied to finding words by image matching, that is to say that the process locates user-specified words in the document images. Through this method the performance of the optical character recognition systems deteriorate severely when confronted with degraded images at the presence of image noise and poor printing quality and when adjacent characters are joined or fused. This is because these systems rely on the segmentation procedure. A segmentation-free approach using Hausdorff distance measure is developed which yields promising results for word image matching [62].

As the error measurement between the target curve and approximation curve, the Hausdorff distance is used in CAD/CAM or Approximation Theory. Hausdorff distance between offset of quadratic Bezier curve and its quadratic approximation is calculated for error measurement [63].

A method is examined that automatically detects and attributes neuroanatomical names to the substructures of the cortical folds using image analysis methods applied to magnetic resonance data of human brains. For label assignment process the Hausdorff distance metric is used as shape similarity measure. Correlations between substructures of brain labellings and functional activations measured by fMRI studies are left as future work [64].

As a result of this chapter, blind deconvolution is an inspiring method for fMRI analysis, so we decided to blindly estimate HRF from the given fMRI signal without any knowledge about the underlying stimulus pattern. For this purpose we developed MAP Blind Deconvolution method which will be detailed mathematically in the following chapter. Also, by adjusting the parameters of the convolution filter we have created a hemodynamic response time series which includes the hemodynamical peaks at the stimulus locations and along their durations so that we do not loose the effect of underlying stimulus pattern. Then we used these HRF time series as inputs for clustering the voxels as active and inactive (passive and clusters due to other artifacts). We used a standard spectral clustering algorithm which can detect the intrinsic geometry within data. Being aware of the selection of the proper metric is crucial for the success of the clustering algorithm; we tried a number of common metrics and a special metric that was developed for fMRI analysis within spectral clustering and finally we decided that Hausdorff distance gives most promising results for our approach. In the following chapter the comparisons and discussions about the clustering metrics will be given and modified Hausdorff distance will be explained in details.

CHAPTER 3

3 METHODOLOGY

3.1 Maximum a Posteriori (MAP) Blind Deconvolution of fMRI

For most cases, fMRI signals are known to suffer from low SNR due to several subject or hardware dependent conditions. To increase the SNR, we estimate hemodynamic response function (HRF), which is known as the impulse response of brain voxels under a neural task. Also we extract a time series signal including HRF's, which is basically the change in the fMRI magnitudes triggered by the neural changes due to the given stimulus pattern. Hemodynamic response is essential for a better understanding of neural activity and it needs to be obtained properly in order to use the information provided by the fMRI signal for drawing conclusions about the underlying unobserved neuronal activation.

The observed time series, fMRI signal in this application, is modeled as the convolution of two independent signals: hemodynamic response and a convolution filter that in our problem will be used to estimate the stimulus. Recall that the hemodynamic response function is the response to impulse. Since we aim at unveiling the hemodynamic

response, which is buried within a convolution, a deconvolution technique is required. Because of the statistical variety of HRF from a voxel to another, we use blind deconvolution to take a rather general approach instead of imposing an estimated model of the HRF with the deconvolution process. Since the objective of blind deconvolution is to reconstruct the original signal from a linearly convolved measurement without the knowledge of either the original signal or the degradation process (stimulus pattern in our approach), and also seeing the fMRI signal as the degraded version of the underlying HRF, this type of deconvolution is found to fit well the HRF extraction problem.

Hence, the problem of HRF estimation is posed and studied within the framework of blind deconvolution in this thesis as in (3.1)

$$r(t) = d(t) \otimes k(t) + n(t) \tag{3.1}$$

where,

r(t): Observed signal, fMRI

d(t): Hemodynamic response function

k(t): Convolution filter (in our problem this is the unknown stimulus pattern)

n(t): Noise

t : Discrete time

The goal of blind deconvolution is to recover or estimate *d(t)* when only the observed signal, *r(t)* is accessible. Unfortunately, with no prior knowledge about *d(t)* and *k(t),* this is an ill-posed problem because there are infinitely many pairs of *(d(t), k(t))* such that (3.1) is satisfied. Considering the Fourier domain representations,

$$\text{Let } G(f) = \Im(d(t) \otimes k(t)) = \Im(d(t))\,\Im(k(t)) \tag{3.2}$$

$$G(f) = D(f)K(f) \tag{3.3}$$

$$D(f) = G(f)K(f)^{-1} \tag{3.4}$$

Basically, for a given fixed *G(f),* then for every different *K(f),* there exists another pair *(D(f), K(f))= (G(f) K(f)$^{-1}$,K(f))* such that it satisfies (3.2), since Fourier Transform takes values from the complex domain in which every element has a multiplicative inverse. This non-uniqueness of solution makes blind deconvolution, in its most general form, ill-posed. And actually that is why it is named 'blind'; no prior knowledge exists on *d(t)* and *k(t)*.

Recent researches show that a key success in blind deconvolution algorithms is to consider the overall shape of the posterior distribution. There is a large amount of research [83], [84], [85], [86], [87], [88], [89] that studies the blind deconvolution problem within a Bayesian framework through Maximum A Posteriori (MAP) estimation. All of those works report favorable results over other types of blind deconvolution techniques. Therefore, we also follow the MAP approach for HRF estimation in this thesis. In an iterative MAP estimation, one tries to find the most likely estimate for the convolution filter (stimulus pattern in our case) given the target signal (HRF in our case) and for the target signal given the convolution filter. This is, in general, expressed as a maximization over the probability distribution of the posterior using Bayes' rule. Mathematically,

$$(d^*(t), k^*(t)) = \arg\max_{d(t),k(t)} p(d(t), k(t) \mid r(t)) \tag{3.5}$$

If one treats *d(t)* and *k(t)* as some parameters of the posterior distribution, then this maximization turns out to be the Maximum Likelihood Estimation. On the other hand, if

we assume that a prior distribution exists over *d(t)* and *k(t)*, then *d(t)* and *k(t)* can be treated as random variables as in Bayesian Statistics. Then the posterior distribution becomes as in (3.6):

$$\begin{aligned}(d^*(t),k^*(t)) &= \arg\max_{d(t),k(t)} p(d(t),k(t)\mid r(t)) \\ &= \arg\max_{d(t),k(t)} \frac{p\left(r(t)\mid d(t),k(t)\right)p\left(d(t),k(t)\right)}{p\left(r(t)\right)}\end{aligned} \tag{3.6}$$

In all blind deconvolution problems one has to take some application dependent assumptions in order to cope with the ill-posed nature of the problem. MAP approach mathematically formulates these assumptions as some prior distributions on the convolution filter, target signal, and the noise in the blind deconvolution problem. This allows one to treat the problem in the framework of Bayesian Statistics. In what follows, similarly, we try to develop this prior knowledge for fMRI data analysis.

In general, fMRI data analysis is based on the observation that the local energy consumption in the brain correlates with the underlying neural activity. Although the exact coupling of neural activation to the vascular system is unknown, the mechanism of this coupling generates significant blurring and delays to the original responses over time, which suggests a low-pass filtering operation. Hemodynamic events have a time scale of few seconds, whereas neural events happen almost instantly [72]. Furthermore, due to the finite sampling of brain volume as voxels, hemodynamic response from a single voxel is thought to be the average of all responses in the space of that voxel [73]. Because of the slow responsiveness of hemodynamics to the neural activation and because HRF is likely to be the average of all such responses within the space of a voxel, we impose, in our problem at hand, ‘smoothness’ on the hemodynamic response, *d(t)*. That is to say, we do not expect to see sudden changes or jumps in the hemodynamics. Although such sudden changes or jumps do exist in the raw fMRI, we model them as noise. The ‘smoothness of the hemodynamics’ is measured as the sum of

all derivative squares in a hemodynamic response. As that value decreases, the hemodynamic response is said to be smoother. In the limiting case, if the value tends to zero, then basically we have a flat hemodynamic response. Mathematically, smoothness expressed in discrete time is thus:

given a hemodynamic response function *d(t):*

$$smoothness\,(d(t)) = \sum_{i=0}^{N-2} (d(i) - d(i+1))^2 + (d(N-1) - d(N-2))^2 \tag{3.7}$$

Thus, using vector notation,

$$smoothness(\mathbf{d}) = \left\|\mathbf{L}\mathbf{d}\right\|^2 \tag{3.8}$$

where,

$$\begin{aligned} &\mathbf{L}\,\mathbf{d}(i) = \mathbf{d}(i) - \mathbf{d}(i+1) \qquad\qquad i = 0,\ldots N-2 \\ &\mathbf{L}\,\mathbf{d}(N-1) = \mathbf{d}(N-1) - \mathbf{d}(N-2) \end{aligned} \tag{3.9}$$

Then (3.7) becomes:

$$\begin{aligned} smoothness\,(d(t)) &= \sum_{i=0}^{N-2} (d(i) - d(i+1))^2 + (d(N-1) - d(N-2))^2 \\ &= \left\|\mathbf{L}\mathbf{d}\right\|^2 = (\mathbf{L}\mathbf{d})^T(\mathbf{L}\mathbf{d}) = \mathbf{d}^T\mathbf{L}^T\mathbf{L}\mathbf{d} \end{aligned} \tag{3.10}$$

$$\mathbf{L}=\begin{bmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \\ & & & -1 & 1 \end{bmatrix}_{NxN}$$ is the difference operator approximating the derivatives via the first order differences.

$$\mathbf{d}=\begin{bmatrix} d(0) \\ \vdots \\ d(i) \\ \vdots \\ d(N-1) \end{bmatrix}_{\mathrm{Nx1}}$$ is the vector form of the signal hemodynamic response.

In our model, we want to impose smoothness on hemodynamic response using the minimization of this smoothness metric:

$$\min_{\mathbf{d}} smoothness(\mathbf{d}) = \min \mathbf{d}^T \mathbf{L}^T \mathbf{L}\mathbf{d} \tag{3.11}$$

Note that, as *smoothness*(**d**) goes to 0, then

$$\begin{aligned} &\|\mathbf{L}\mathbf{d}\|^2 \to 0 \Rightarrow \sum_{i=0}^{N-2} (d(i)-d(i+1))^2 + (d(N-1)-d(N-2))^2 \to 0 \Rightarrow \\ &\forall i, d(i)-d(i+1) \to 0 \Rightarrow \mathbf{d} \to \mathbf{constant}\, such\, that\, \forall i, \mathbf{constant}(i) = c \end{aligned} \tag{3.12}$$

Then, we obtain a flat signal with zero derivative everywhere which is the most smooth signal. This is definitely we do not want to end up with and need some deeper consideration. We will explain this later and for now, let us continue analyzing our issue of smoothness.

Probabilistically, enforcing smoothness on the hemodynamic response through the minimization in (3.11) is nothing but putting a Gaussian prior distribution on**d**. For the sake of formulating our smoothness idea within the framework of Bayesian Statistics, now, we show how the minimization in (3.11) can be re-posed as a maximization over a Gaussian distribution.

Since the exponential function e^x is a monotonically increasing function, if we take the exponential of the 'smoothness', we obtain an equivalent minimization in terms of the argument:

$$\begin{aligned} \arg\min_d \textit{smoothness}\ (\mathbf{d}) &= \arg\min_d \mathbf{d}^T\mathbf{L}^T\mathbf{L}\mathbf{d} \\ &= \arg\min_d e^{\mathbf{d}^T\mathbf{L}^T\mathbf{L}\mathbf{d}} \end{aligned} \tag{3.13}$$

Taking the negative of the exponent together with an insignificant constant C_d, then the minimization turns into maximization,

$$\begin{aligned} \arg\min_{\mathbf{d}} \textit{smoothness}\ (\mathbf{d}) &= \arg\min_d e^{\mathbf{d}^T\mathbf{L}^T\mathbf{L}\mathbf{d}} \\ &= \arg\max_d C_d e^{-\mathbf{d}^T\mathbf{L}^T\mathbf{L}\mathbf{d}} \end{aligned} \tag{3.14}$$

Then also we can multiply the exponent with a positive real number $\frac{1}{2\lambda^2}$;

$$\arg\min_{\mathbf{d}} smoothness\,(\mathbf{d}) = \arg\max_{d} C_d e^{\frac{-\mathbf{d}^T\mathbf{L}^T\mathbf{L}\mathbf{d}}{2\lambda^2}} \tag{3.15}$$

Let $S(i,j) = \frac{L(i,j)}{\lambda}$ then,

$$\begin{aligned}\arg\min_{d} smoothness\ \ (\mathbf{d}) &= \arg\max_{d} C_d e^{\frac{-\mathbf{d}^T\mathbf{L}^T\mathbf{L}\mathbf{d}}{2\lambda^2}} \\ &= \arg\max_{d} C_d e^{\frac{-\mathbf{d}^T\mathbf{S}^T\mathbf{S}\mathbf{d}}{2}}\end{aligned} \tag{3.16}$$

Considering the difference matrix **L**, its last row is just the opposite sign of its *(N-1)'*th row, i.e, *L*(*N, :*) = - *L*(*N-1, :*), (MATLAB notation)

Hence its determinant is zero; however, we can do a benign trick on this. Since in general the fMRI time series is long in the order of hundreds, we can discard the derivative at the last time instant, and we can adjust our difference matrix **L** in the following way:

$$\mathbf{L} = \begin{bmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \\ & & & 0 & 0.001 \end{bmatrix}_{N \times N}$$

By adjusting the last row of *L* as above, instead of calculating the difference on **d**(N) directly, we assume it normalized $\frac{\mathbf{d}(N)}{1000}$, because the mean of the first order differences

of fMRI that we study is approximately 1/1000 of the mean intensity. This normalization facilitates our calculations, since L becomes invertible: det(**L**)=0.001 (independent of N).

Now in this new form, since **L** invertible, then $\mathbf{L}^T\mathbf{L}$ is a positive definite matrix.

Proof:

For every nonzero x,

$$\mathbf{x}^T\mathbf{L}^T\mathbf{L}\mathbf{x} = (\mathbf{L}\mathbf{x})^T(\mathbf{L}\mathbf{x}) = \|\mathbf{L}\mathbf{x}\|^2 \geq 0 \tag{3.17}$$

Since **L** is invertible, then $\text{NULL}(\mathbf{L}) = \phi$ and then since x is nonzero $\mathbf{L}\,\mathbf{x} \neq 0$, hence;

$$\mathbf{x}^T\mathbf{L}^T\mathbf{L}\mathbf{x} = (\mathbf{L}\mathbf{x})^T(\mathbf{L}\mathbf{x}) = \|\mathbf{L}\mathbf{x}\|^2 > 0 \tag{3.18}$$

$\mathbf{L}^T\mathbf{L}$ is strictly positive definite.

Then $\mathbf{S}^T\mathbf{S}$ is also strictly positive definite and its inverse exists. Let $\Sigma = (\mathbf{S}^T\mathbf{S})^{-1}$

As a result of this,

$$\begin{aligned}
\arg\min_{\mathbf{d}} \textit{smoothness}\ (\mathbf{d}) &= \arg\max_d C_d e^{\frac{-\mathbf{d}^T\mathbf{L}^T\mathbf{L}\mathbf{d}}{2\lambda^2}} \\
&= \arg\max_d C_d e^{\frac{-\mathbf{d}^T\mathbf{S}^T\mathbf{S}\mathbf{d}}{2}} \\
&= \arg\max_d C_d e^{\frac{-\mathbf{d}^T\,\Sigma^{-1}\,\mathbf{d}}{2}}
\end{aligned} \tag{3.19}$$

Note that, with a normalization constant C_d $p(\mathbf{d}) = C_d e^{\frac{-\mathbf{d}^T \Sigma^{-1} \mathbf{d}}{2}}$ is Gaussian with zero mean and Σ covariance. Then basically,

$$\begin{aligned} \arg\min_{\mathbf{d}} smoothness(\mathbf{d}) &= \arg\max_{\mathbf{d}} C_d e^{\frac{-\mathbf{d}^T \mathbf{L}^T \mathbf{L} \mathbf{d}}{2\lambda^2}} \\ &= \arg\max_{\mathbf{d}} C_d e^{\frac{-\mathbf{d}^T \mathbf{S}^T \mathbf{S} \mathbf{d}}{2}} \\ &= \arg\max_{\mathbf{d}} C_d e^{\frac{-\mathbf{d}^T \Sigma^{-1} \mathbf{d}}{2}} \\ &= \arg\max_{\mathbf{d}} p(\mathbf{d}) \end{aligned} \tag{3.20}$$

Smoothness constraint on the hemodynamic response implies, in our model, the minimization of the magnitude of the square sum of derivatives and this turns out to be a Gaussian prior, which favors or assigns high probabilities to low derivative magnitudes. And note that in the limiting case, this prior assigns the maximum probability to the zero, mean vector (in general, mode of a Gaussian is equal to the mean) that is a flat signal, the smoothest signal.

Our second prior is on the noise signal in our model. Since we bury the high frequency noises in our fMRI measurement process in this noise component, we assume it to be Gaussian and independent at every time instant, which is, by definition Additive White Gaussian Noise (AWGN). Since it is independent at every time instant, it has a diagonal covariance with σ^2 at diagonal entries.

$$p(\mathbf{n}) = C_n e^{\frac{-\|\mathbf{n}\|^2}{2\sigma^2}} \tag{3.21}$$

Third, we put a uniform prior on the convolution filter **k**. That is to say, we do not have a constraint on the shape or magnitudes of the stimulus pattern except that we assume it to be finite impulse response of some length *p* with $\forall i,\ \mathbf{k}(i) > 0$.

$$p(\mathbf{k}) = C_k \text{ such that } \forall i,\ \mathbf{k}(i) > 0 \tag{3.22}$$

Here the length of the convolution filter (stimulus) *p* has an important meaning. There are basically two different cases resulting from p: (1) If the convolution filter has a small length with respect to the full length N, then we hope to recover one impulse block (a single square; we think of the underlying stimulus as a square wave) of the underlying stimulus pattern in the fMRI application as the convolution filter. In this case, then, the other component of the convolution *d(t)* will be a time series signal including a series of hemodynamics. We name it 'hemodynamical time series' (2) Secondly, if the convolution filter is of full length N, or close to being full length, then we hope to recover the whole stimulus pattern (whole square wave) as our convolution filter. Then in this case, *d(t)* will be the impulse function, in other words for our approach, the hemodynamic response function (HRF). In this thesis, we use hemodynamical time series for clustering of fMRI time series and HRF extraction is studied separately. The reason behind not using HRF in clustering is that: when HRF is extracted the stimulus is separated and captured as *k(t)*. On the other hand, hemodynamical time series clearly includes the information of the stimulus in it such as the location of impulses as well as the duration of them. Since the impulse is the only separation regarding the activation detection, we use hemodynamical time series in our clustering. These will be discussed both in this chapter as well as the experiments chapter.

Lastly, we also put a uniform prior on the observations meaning that we have no constraints for the observations.

$$p(\mathbf{r}) = C_r \tag{3.23}$$

Having constructed all of our priors for the components of our convolution model, we can repose our problem and goal:

Problem:

$$r(t) = d(t) \otimes k(t) + n(t) \tag{3.24}$$

where,

r(t): Observed signal, fMRI: $p(\mathbf{r}) = C_r$

d(t): Hemodynamic response function: $p(\mathbf{d}) = C_d e \frac{-\mathbf{d}^T \sum^{-1} \mathbf{d}}{2}$

k(t): Finite Impulse Response (FIR) convolution filter: $p(\mathbf{k}) = C_k$ such that $\forall i$, $\mathbf{k}(i) > 0$

n(t): Additive White Gaussian Noise (AWGN): $p(n) = C_n e^{\frac{-\|\mathbf{n}\|^2}{2\sigma^2}}$

t : Discrete time

Goal:

Given the observations, *r(t)*,

Estimate *d(t)*, *k(t)*, *n(t)* subject to the priors.

Solution (1):

For blind deconvolution problems, recent research in [77] shows that to consider the overall shape of posterior distribution $p(\mathbf{k},\mathbf{d}|\mathbf{r})$ provides promising results. We take a MAP approach in this work in accordance with the latest research activities in the literature. Since Log is a monotone increasing function, it would not affect the maximization of the posterior distribution however it facilitates the derivations by letting us avoid the exponentials. Together with a negative sign, it becomes minimization:

$$
\begin{aligned}
(\mathbf{d}^*, \mathbf{k}^*) &= \arg\min_{(\mathbf{d},\mathbf{k})} -\log p(\mathbf{k},\mathbf{d}|\mathbf{r}) \\
&= \arg\min_{(\mathbf{d},\mathbf{k})} -\log \left\{ \frac{p(\mathbf{r}|\mathbf{d},\mathbf{k})\, p(\mathbf{d})\, p(\mathbf{k})}{p(\mathbf{r})} \right\} \qquad (3.25) \\
&= \arg\min_{(\mathbf{d},\mathbf{k})} -\log \left\{ p(\mathbf{r}|\mathbf{d},\mathbf{k})\, p(\mathbf{d})\, p(\mathbf{k}) \right\}
\end{aligned}
$$

Note that we assume **k** and **d** are independent since we do not expect any functional dependencies in between them, and p(**r**) does not show in (3.25) since it is constant.
In the following derivations we always have the constraint: $\forall i,\ \mathbf{k}(i) > 0$

Then,

$$(\mathbf{d}^*, \mathbf{k}^*) = \arg\min_{(\mathbf{d},\mathbf{k})} -\log\left\{p(\mathbf{r}|\mathbf{d},\mathbf{k})\, p(\mathbf{d})\, p(\mathbf{k})\right\}$$

$$= \arg\min_{(\mathbf{d},\mathbf{k})} -\log\left\{C_n e^{\frac{-\left\| r(t)-d(t)\otimes k(t)\right\|^2}{2\sigma^2}} C_d e^{\frac{-\mathbf{d}^T \sum^{-1} \mathbf{d}}{2}} C_k\right\}$$

$$= \arg\min_{(\mathbf{d},\mathbf{k})} -\left\{\log C_n + \log C_d + \log C_k - \frac{\left\| r(t) - d(t)\otimes k(t)\right\|^2}{2\sigma^2} - \frac{\mathbf{d}^T \sum^{-1} \mathbf{d}}{2}\right\}$$

$$= \arg\min_{(\mathbf{d},\mathbf{k})} \left\{\frac{\left\| r(t) - d(t)\otimes k(t)\right\|^2}{2\sigma^2} + \frac{\mathbf{d}^T \sum^{-1} \mathbf{d}}{2}\right\}$$

$$= \arg\min_{(\mathbf{d},\mathbf{k})} \left\{\frac{\left\| r(t) - d(t)\otimes k(t)\right\|^2}{2\sigma^2} + \frac{\mathbf{d}^T \mathbf{L}^T \mathbf{L} \mathbf{d}}{2\lambda^2}\right\} \qquad (3.26)$$

Here we can write the first term in vector notation using $\mathbf{A_k}$ which is:

$$\mathbf{A_k} = \begin{bmatrix} k(0) & & & & & & \\ k(1) & k(0) & & & & & \\ \vdots & \ddots & \ddots & & & & \\ k(p-1) & \cdots & k(1) & k(0) & & & \\ 0 & k(p-1) & \cdots & k(1) & k(0) & \cdots & 0 \\ & & & & \ddots & \ddots & \\ & & & & & k(1) & k(0) \end{bmatrix}_{\text{N x N}}$$

$\mathbf{A_k}$ is the convolution matrix of *k(t)* such that;

$$d(t) \otimes k(t) = \mathbf{A}_{\mathbf{k}}\mathbf{d} = \sum_{w=0}^{\text{N-1}} d(w)\, k(t-w) \tag{3.27}$$

Hence our first attempt MAP approach suggests the optimization problem in (3.28) for a solution to our blind deconvolution problem:

$$\begin{aligned}
(\mathbf{d}^*, \mathbf{k}^*) &= \arg\min_{(\mathbf{d},\mathbf{k})} -\log\left\{p(\mathbf{r}|\mathbf{d},\mathbf{k})\, p(\mathbf{d})\, p(\mathbf{k})\right\} \\
&= \arg\min_{(\mathbf{d},\mathbf{k})} \left\{ \frac{\left\| r(t) - d(t) \otimes k(t) \right\|^2}{2\sigma^2} + \frac{\mathbf{d}^T \mathbf{L}^T \mathbf{L} \mathbf{d}}{2\lambda^2} \right\} \\
&= \arg\min_{(\mathbf{d},\mathbf{k})} \left\{ \frac{\left\| \mathbf{r} - \mathbf{A}_{\mathbf{k}}\mathbf{d} \right\|^2}{2\sigma^2} + \frac{\mathbf{d}^T \mathbf{L}^T \mathbf{L} \mathbf{d}}{2\lambda^2} \right\} \\
&= \arg\min_{(\mathbf{d},\mathbf{k})} \left\{ \kappa \left\| \mathbf{r} - \mathbf{A}_{\mathbf{k}}\mathbf{d} \right\|^2 + \mathbf{d}^T \mathbf{L}^T \mathbf{L} \mathbf{d} \right\} \qquad \text{where } \kappa = \frac{\lambda^2}{\sigma^2}
\end{aligned} \tag{3.28}$$

This optimization problem basically tries to approximate the observed signal as a convolution of **d** and **k** while at the same time having the hemodynamic **d** as smooth as possible. And the parameter κ nicely controls the trade-off between the approximation of the observations and the smoothness of the hemodynamics. Unfortunately, this solution is badly problematic because of the following reason:

Let us pick a pair (**d, k**) and consider another pair (**d', k'**) such that **d'**=s**d**, and **k'**=(1/s)**k**, where s is a scaling factor.

As $s \to 0$;

(1) The first term of the optimization in (3.28) stays the same since

$$\left\| r(t) - d(t) \otimes k(t) \right\|^2 = \left\| r(t) - d'(t) \otimes k'(t) \right\|^2 \tag{3.29}$$

(2) However, the second term that stands for the smoothness of the hemodynamic decreases, overall the optimization improves!

Therefore, the joint optimization always favors to flat signal for hemodynamic since as $s \to 0$ then $d' \to 0$ as well. To tackle this problem, we modify the solution above using an iterative optimization of the same cost function through the Expectation-Maximization Algorithm (EM) by basically alternating between the optimum hemodynamic given the convolution filter and the optimum convolution filter given the hemodynamic. With the help of an EM type iterative algorithm, therefore, we can avoid the joint optimization and get a suboptimal solution for the same optimization problem avoiding the flat hemodynamic problem. This algorithm can be summarized as follows:

Solution (2): Iterative EM optimization for MAP estimate of hemodynamic

- Input:
  - Observations, $\mathbf{r}$
  - Initial estimate for convolution filter, $\mathbf{k}_0$
  - Maximum Iteration number, *iter*.
  - Parameters: p, κ
- Let $\mathbf{k} \leftarrow \mathbf{k}_0, i \leftarrow 1,$
- While $i<iter$
  - EM optimization:
    - E-step
      - $\mathbf{d} \leftarrow E_{\mathbf{d}}\{\mathbf{d} \mid \mathbf{k}, \mathbf{r}\}$
    - M-step (Likelihood Maximization)
      - $\mathbf{k} \leftarrow \arg\max_{\mathbf{k}} p(\mathbf{k} \mid \mathbf{d}, \mathbf{r})$
    - $i \leftarrow i+1$
  - return $(\mathbf{k}, \mathbf{d})$

In the following, we give the details of our algorithm:

E-step

In this step the algorithm calculates the mean hemodynamic, $E_{\mathbf{d}}(\mathbf{d} \mid \mathbf{k}, \mathbf{r})$, given the convolution filter and the observations.

Note that:

$$p(\mathbf{d}, \mathbf{r} \mid \mathbf{k}) = p(\mathbf{r} \mid \mathbf{d}, \mathbf{k}) p(\mathbf{d} \mid \mathbf{k}) = C_n \, e^{\frac{-\| \mathbf{r} - \mathbf{A_k d} \|^2}{2\sigma^2}} \, C_d \, e^{\frac{-\mathbf{d}^{\mathbf{T}} \mathbf{L}^{\mathbf{T}} \mathbf{L} \mathbf{d}}{2\lambda^2}} \tag{3.30}$$

and since **r** is also known:

$$p(\mathbf{d} \mid \mathbf{r}, \mathbf{k}) = C \, C_n \, e^{\frac{-\| \mathbf{r} - \mathbf{A_k d} \|^2}{2\sigma^2}} \, C_d \, e^{\frac{-\mathbf{d}^{\mathbf{T}} \mathbf{L}^{\mathbf{T}} \mathbf{L} \mathbf{d}}{2\lambda^2}} \tag{3.31}$$

where C is just a normalization constant such that integral of $p(\mathbf{d} \mid \mathbf{r}, \mathbf{k})$ is 1.

Seeing that $p(\mathbf{d} \mid \mathbf{r}, \mathbf{k})$ is a multiplication of two Gaussian functions, itself must be another Gaussian of which we are seeking the mean, $E_{\mathbf{d}}(\mathbf{d} \mid \mathbf{k}, \mathbf{r})$. Because mean of a Gaussian is also the mode of it, then (3.32) must hold:

$$E_{\mathbf{d}}(\mathbf{d} \mid \mathbf{r}, \mathbf{k}) = \arg\max_{\mathbf{d}} C \, C_n \, e^{\frac{-\| \mathbf{r} - \mathbf{A_k d} \|^2}{2\sigma^2}} \, C_d \, e^{\frac{-\mathbf{d}^{\mathbf{T}} \mathbf{L}^{\mathbf{T}} \mathbf{L} \mathbf{d}}{2\lambda^2}} \tag{3.32}$$

The derivations we already made in (3.25), (3.26), (3.27) and (3.28) simplify (3.32) into:

$$E_{\mathbf{d}}(\mathbf{d}|\mathbf{r},\mathbf{k}) = \arg\max_{\mathbf{d}} C\, C_n\, e^{\frac{-\|\mathbf{r}-\mathbf{A}_{\mathbf{k}}\mathbf{d}\|^2}{2\sigma^2}}\, C_d\, e^{\frac{-\mathbf{d}^{\mathbf{T}}\mathbf{L}^{\mathbf{T}}\mathbf{L}\mathbf{d}}{2\lambda^2}}$$
$$= \arg\min_{\mathbf{d}} \kappa\|\mathbf{r}-\mathbf{A}_{\mathbf{k}}\mathbf{d}\|^2 + \mathbf{d}^{\mathbf{T}}\mathbf{L}^{\mathbf{T}}\mathbf{L}\mathbf{d}; \qquad \text{where } \kappa = \frac{\lambda^2}{\sigma^2} \tag{3.33}$$

The λ parameter in the optimization (3.32) controls the covariance of the prior on p(**d**). As it increases, we start allowing larger derivatives, and so tolerating less smooth hemodynamics, on the other hand, smaller λ implies a tighter prior around the mean, 0, which in turn makes our model favor smooth hemodynamics more. In the limiting cases, if λ goes to infinity, note that the second term of the cost which stands for the smoothness condition of the hemodynamic goes to zero and all of the cost gets concentrated on the first term which is the data fitting term. If λ goes to 0, then the second term goes to infinity and all of the cost this time gets concentrated on the smoothness constraint. As for the σ parameter, it controls the fitting accuracy of the convolution **d** with **k** to the observations. Larger σ means a larger variance for the noise in the model which then allows relatively a larger deviation between the observations and the convolution. On the other hand if the noise variance σ is small, then it means we do not really expect much noise and so we do want a tighter fitting to the observations by the convolution. Similarly, if σ goes to 0, data fitting term gets dominant and if it goes to infinity then the smoothness constraint gets dominant. For the ease of demonstration, we lump the effect of these two parameters into a single parameter κ of which tuning allows us to we simply adjust the trade-off between the data fitting accuracy of the convolution and the smoothness constraint on the estimated hemodynamic. Hence the minimization we are considering is based on the cost function in (3.34):

$$\cos t\_Estep = \kappa \left\| \mathbf{r} - \mathbf{A_k d} \right\|^2 + \mathbf{d}^T \mathbf{L}^T \mathbf{L d} \tag{3.34}$$

The first term in the cost function is a *least square approximation* problem in which one basically fits a linear function of **d** to the observations **r**. It is a convex optimization problem [78] with an exact solution $\mathbf{d}^* = \mathbf{A}^+ \mathbf{r}$ where $\mathbf{A}^+$ is the pseudo inverse of **A**. This is the situation for the first term. As for the second term this situation is not different, seeing that $\mathbf{d}^T \mathbf{L}^T \mathbf{L d} = \left\| \mathbf{0} - \mathbf{L d} \right\|^2$ is just another *least square approximation* problem which is also convex. Since sum of two convex functions is also convex (closeness of convexity under addition), overall cost of the E-step is convex. Then we can easily calculate the minimum analytically for this optimization by just taking the zero derivative:

$$\frac{\partial}{\partial \mathbf{d}} \left\{ \kappa \left\| \mathbf{r} - \mathbf{A_k d} \right\|^2 + \mathbf{d}^T \mathbf{L}^T \mathbf{L d} \right\} = \frac{\partial}{\partial \mathbf{d}} \left\{ \kappa\, \mathbf{r}^T \mathbf{r} + \kappa\, \mathbf{d}^T \mathbf{A_k}^T \mathbf{A_k d} - 2\kappa\, \mathbf{r}^T \mathbf{A_k d} + \mathbf{d}^T \mathbf{L}^T \mathbf{L d} \right\}$$

$$= 2\kappa\, \mathbf{A_k}^T \mathbf{A_k d} - 2\kappa\, \mathbf{A_k}^T \mathbf{r} + 2\mathbf{L}^T \mathbf{L d} \tag{3.35}$$

$$= 2(\kappa\, \mathbf{A_k}^T \mathbf{A_k} + \mathbf{L}^T \mathbf{L})\mathbf{d} - 2\kappa\, \mathbf{A_k}^T \mathbf{r}$$

Then, solving the partial derivatives for **0**, (Note that, $\left(\kappa\, \mathbf{A_k}^T \mathbf{A_k} + \mathbf{L}^T \mathbf{L}\right)^{-1}$ exists since it is positive definite).

$$\frac{\partial}{\partial \mathbf{d}} \cos t\_Estep = 0$$

$$2(\kappa\, \mathbf{A_k}^T \mathbf{A_k} + \mathbf{L}^T \mathbf{L})\mathbf{d} - 2\kappa\, \mathbf{A_k}^T \mathbf{r} = 0 \tag{3.36}$$

$$\mathbf{d}^* = \kappa \left(\kappa\, \mathbf{A_k}^T \mathbf{A_k} + \mathbf{L}^T \mathbf{L}\right)^{-1} \mathbf{A_k}^T \mathbf{r}$$

$\mathbf{d}^*$ is the optimum MAP solution for the hemodynamic response for a given convolution filter $_{\kappa}$ calculated by the E-step of our algorithm. Note that, this solution is not the trivial solution (which is the flat signal) of the joint optimization we explained as solution(1) simply because $\mathbf{d}^*$ can be $\mathbf{0}$ only when κ=0. Since solution(1) performs a joint optimization over (**d,k**), the degree of freedom is more than the one of our proposed solution(2), as a result, the problem becomes ill-posed as already explained. We tackle this problem with this EM type iterative approach by alternating between the optimum hemodynamic given the convolution filter and the optimum convolution filter given the hemodynamic and by never letting the algorithm get stuck at a trivial solution. Moreover, the parameter κ nicely incorporates our priors.

M-step

In this step, the algorithm maximizes the data likelihood through the maximization, $\max_{\mathbf{k}} p(\mathbf{k}|\mathbf{d},\mathbf{r})$ given the hemodynamic.

$$\begin{aligned}\mathbf{k}^* &= \arg\max_{\mathbf{k}} p(\mathbf{k}|\mathbf{r},\mathbf{d}) \\ &= \arg\max_{\mathbf{k}} C_n e^{\frac{-\| r(t)-k(t)\otimes d(t)\|}{2\sigma^2}} \text{ subject to } \forall i, \mathbf{k}(i)>0\end{aligned} \tag{3.37}$$

We rewrite this optimization in (3.36) using the vector/matrix notation of the convolution $k(t)\otimes d(t)$ as $\mathbf{A_d k}$ using the convolution matrix derived from **d** as opposed to using $\mathbf{A_k}$ previously. Namely:

$$A_d = \begin{bmatrix} d(0) & 0 & \cdots & 0 \\ d(1) & d(0) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ d(p-1) & d(p-2) & \cdots & d(0) \\ d(p) & d(p-1) & \cdots & d(1) \\ \vdots & \vdots & \vdots & \vdots \\ d(N-1) & d(N-2) & \cdots & d(N-p) \end{bmatrix}$$

$\mathbf{A_d}$ is the *Nxp* convolution matrix of d(t) such that;

$$d(t) \otimes k(t) = \mathbf{A_d k} = \sum_{w=0}^{p-1} k(w) d(t-w) \tag{3.38}$$

Hence, our optimization then turns out:

$$\begin{aligned} \mathbf{k}^* &= \arg\max_{\mathbf{k}} \; p(\mathbf{k}|\mathbf{r},\mathbf{d}) = \arg\min_{\mathbf{k}} \left\|\mathbf{r} - \mathbf{A_d k}\right\| \\ &= \arg\min_{\mathbf{k}} \left\{ \frac{1}{2} \mathbf{k}^{\mathrm{T}} \mathbf{A_d}^{\mathrm{T}} \mathbf{A_d k} - \mathbf{r}^{\mathrm{T}} \mathbf{A_d k} \right\} \\ &= \arg\min_{\mathbf{k}} \left\{ \frac{1}{2} \mathbf{k}^{\mathrm{T}} \mathbf{H k} + \mathbf{f}^{\mathrm{T}} \mathbf{k} \right\} \qquad \text{subject to} \; \forall \mathrm{i}, \; \mathbf{k}(i) > 0 \end{aligned} \tag{3.39}$$

where $\mathbf{H} = \mathbf{A_d}^{\mathrm{T}} \mathbf{A_d}$ and $\mathbf{f} = -\mathbf{A_d}^{\mathrm{T}} \mathbf{r}$

This is a quadratic minimization subject to linear constraints [79], and thus a convex problem that can be solved using quadratic programming. Actually there is a dedicated MATLAB routine for such problems: “quadprog”.

Illustration of the MAP blind deconvolution Algorithm:

We illustrate our MAP blind deconvolution algorithm, and explain the outputs of our algorithm at each step with a real fMRI data (Figure 3.1).

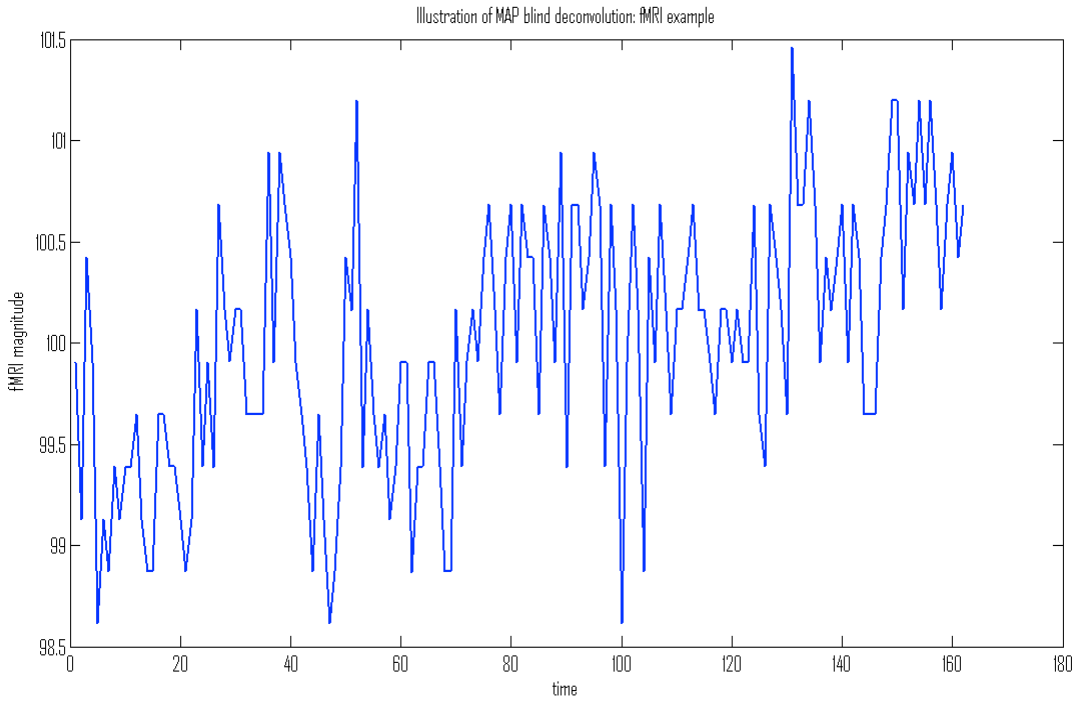


Figure 3.1 An example of real fMRI data

Although there is clearly a large amount of noise in this fMRI signal, the overall behavior of the signal is still detectable. We expect a good hemodynamic estimation to decrease the noise effect and at the same time not to fade away the general trend in the signal. For instance, in the signal of Figure 3.1, we have peaks approximately between the time instants [20, 40], [50, 60], [70, 100], [100 120]. Peak magnitudes (after the noise is estimated and extracted), and their locations are known to be among important features of fMRI which give clues about the underlying neural activity. This is because they are correlated with the stimulus governing that neural activity [70]. For instance in the signal of Figure 3.1, we can predict that there exists stimulus in the aforementioned

intervals with the help of peak magnitudes and their locations. Especially these features are supposed to be extracted by the hemodynamic estimation and all other effects are supposed to be then suppressed as much as possible which in total should increase the SNR.

Setting the parameters κ=0.1 as we want to put more importance on the smoothness of the hemodynamic and p=10 (a small length filter) as we want to extract the hemodynamical time series to preserve the effect of stimulus and still have denoised hemodynamical signal, we are able to nicely observe the hemodynamical peaks as seen in Figure 3.2.

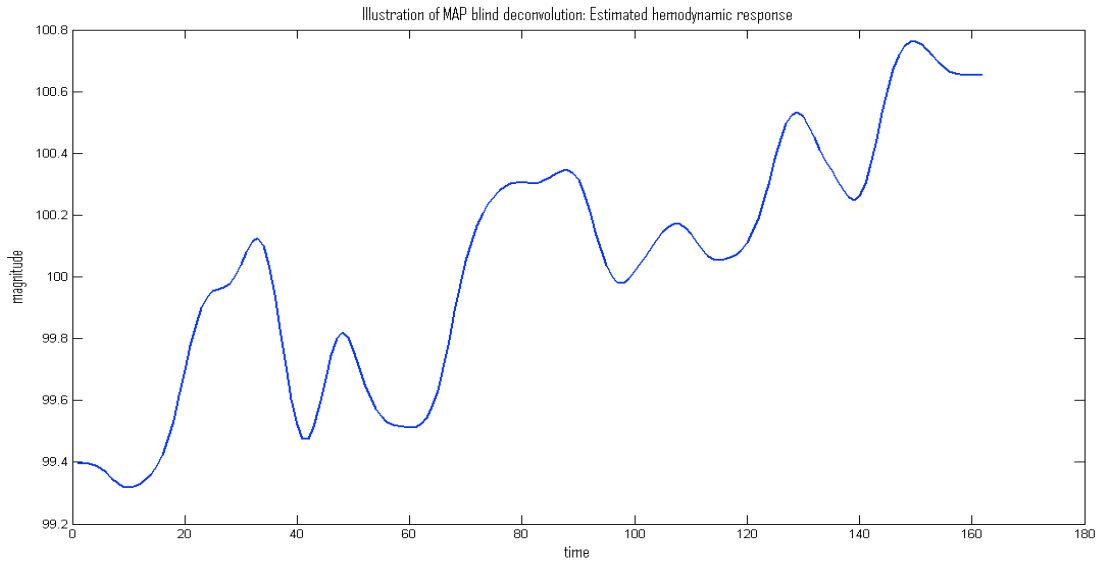


Figure 3.2 Estimated hemodynamic response function

Visually speaking, MAP blind deconvolution successfully eliminates the noise from the fMRI signal of Figure 3.1 and extracts the peak magnitude and location features. Furthermore, it clearly estimates the overall shape. Note that comparing to the signal of Figure 3.1, the estimated hemodynamical time series is clearly cleaned and denoised while it still does have the peaks in the aforementioned intervals. And since it is denoised, now the peaks are far more detectable which effectively means an increase in

SNR in the signal of Figure 3.2 when compared that of Figure 3.1. In general, from a voxel to another, activation strength and patterns can vary significantly even if they belong to active regions in the brain since different regions of the brain depending on their functionality can respond differently to the same neural activity. The interactions between the voxels also add onto this variety. Hence, most of the modeling approaches in the literature, which try to model active or passive hemodynamics uniformly over the voxels, are not capable of considering the variety of hemodynamic responses between voxels. With the help of the blind deconvolution, which is unsupervised (stimulus is assumed unknown) and model-free (no particular shape is assumed for the hemodynamics), we overcome this issue.

Using blind deconvolution generates a model free approach in estimating the hemodynamics only through extracting the good features. This increases significantly the estimation accuracy.

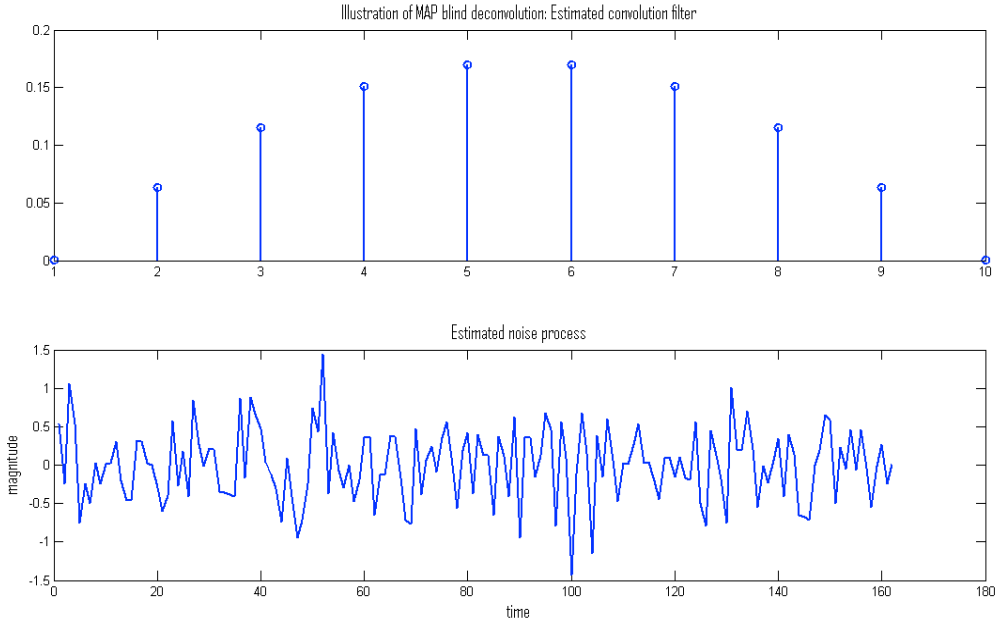


Figure 3.3 Estimated convolution filter and noise process

Figure 3.3 shows an estimated convolution filter when the signal of Figure 3.1 is used in our MAP blind deconvolution. We consider this as one single impulse block of a given square wave like stimulus pattern. And our blind deconvolution locates a series of hemodynamics in *d(t)* each of which passes through the convolution filter in the process of convolution, and generates the fMRI before noise. It has a Gaussian shape centered at 5 or 6 and is decaying on tails which immediately reveals a delay between the observed signal and the hemodynamical time series. This is because in the process of convolution, each hemodynamic of the hemodynamical time series generates when it hits the peak of the convolution filter taking 5 time points. Note that the peak of the hemodynamic corresponding to the interval [40 60] in Figure 3.2 is closer to the time point 35, and on the other hand, in the observed fMRI signal (Figure 3.1) the same peak is closer to the time point 40 which is the mentioned delay. This is actually consistent with what we should expect as delay between the underlying neural activity and the actual time when the brain develops a physical and observable response. Hence, the moment that a new neural response is generated, has an effect on the raw fMRI in a delay which is, in this simple example 5 time points. The noise process is also shown in Figure 3.3. If it is added to the convolution output, we precisely recover back the raw fMRI.

What follows in Figure 3.4 is the model log-likelihood (up to a scaling factor) of the observations against the iterations of our algorithm:

$$model\_log\_likelihood = \log\left\{p(\mathbf{r}|\mathbf{k},\mathbf{d})\,p(\mathbf{d})\,p(\mathbf{k})\right\}$$

$$= -\frac{\left\|\mathbf{r}-\mathbf{A}_{\mathbf{k}}\mathbf{d}\right\|^2}{2\sigma^2} - \frac{\mathbf{d}^{\mathrm{T}}\mathbf{L}^{\mathrm{T}}\mathbf{L}\,\mathbf{d}}{2\lambda^2} \tag{3.40}$$

$$= -\kappa\left\|\mathbf{r}-\mathbf{A}_{\mathbf{k}}\mathbf{d}\right\|^2 - \mathbf{d}^{\mathrm{T}}\mathbf{L}^{\mathrm{T}}\mathbf{L}\,\mathbf{d}; \quad \text{where} \quad \kappa = \frac{\lambda^2}{\sigma^2}$$

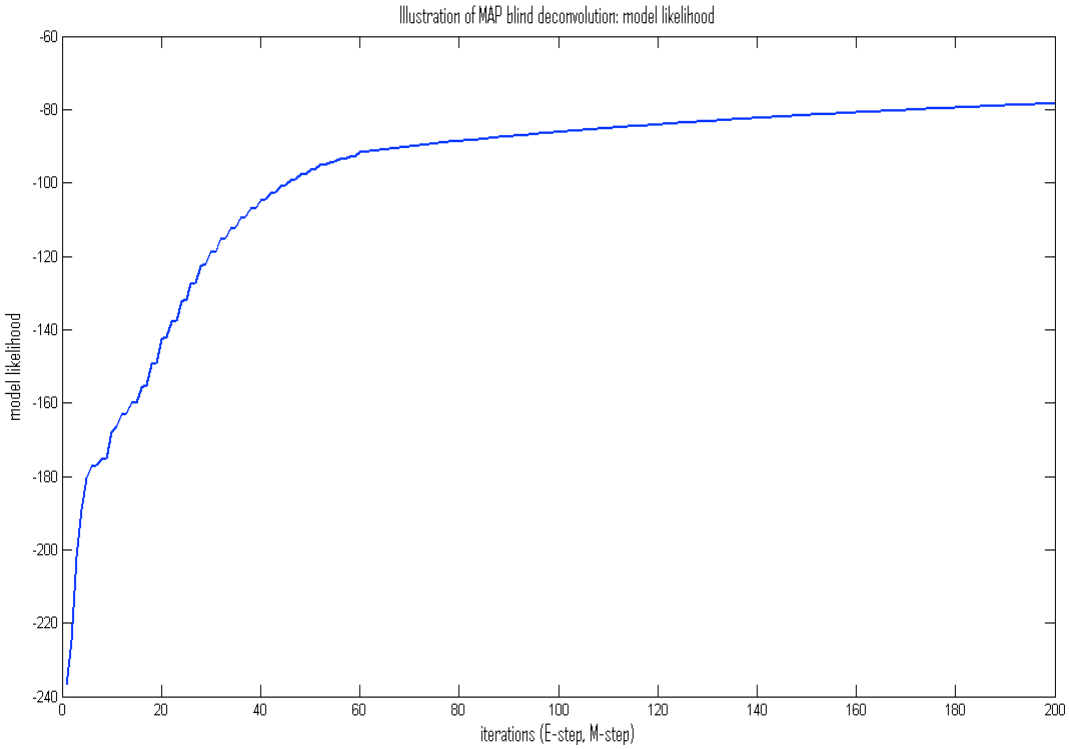


Figure 3.4 Likelihood of the model

As shown in Figure 3.4, our MAP blind deconvolution algorithm successfully increases the likelihood of our model at every iteration. Moreover, the convergence is also quick, after around 30(x2 for E_step and M_step) iterations for the optimization to converge in the estimations.

Figure 3.5 illustrates the effect of the parameter κ on the algorithm when p=10. These plots justify our theoretical explanations. The parameter κ controls the trade-off between the data fitting by the convolution and the smoothness of the estimated hemodynamic. Setting κ=0.1 is reasonable for hemodynamic estimation since it seems to yield a balanced estimation regarding the noise reduction and extraction good features. Note that when κ=0.5 or larger; the estimated hemodynamical time series become noisy. And when κ=0.05 the peak of the interval [100 120] fades away.

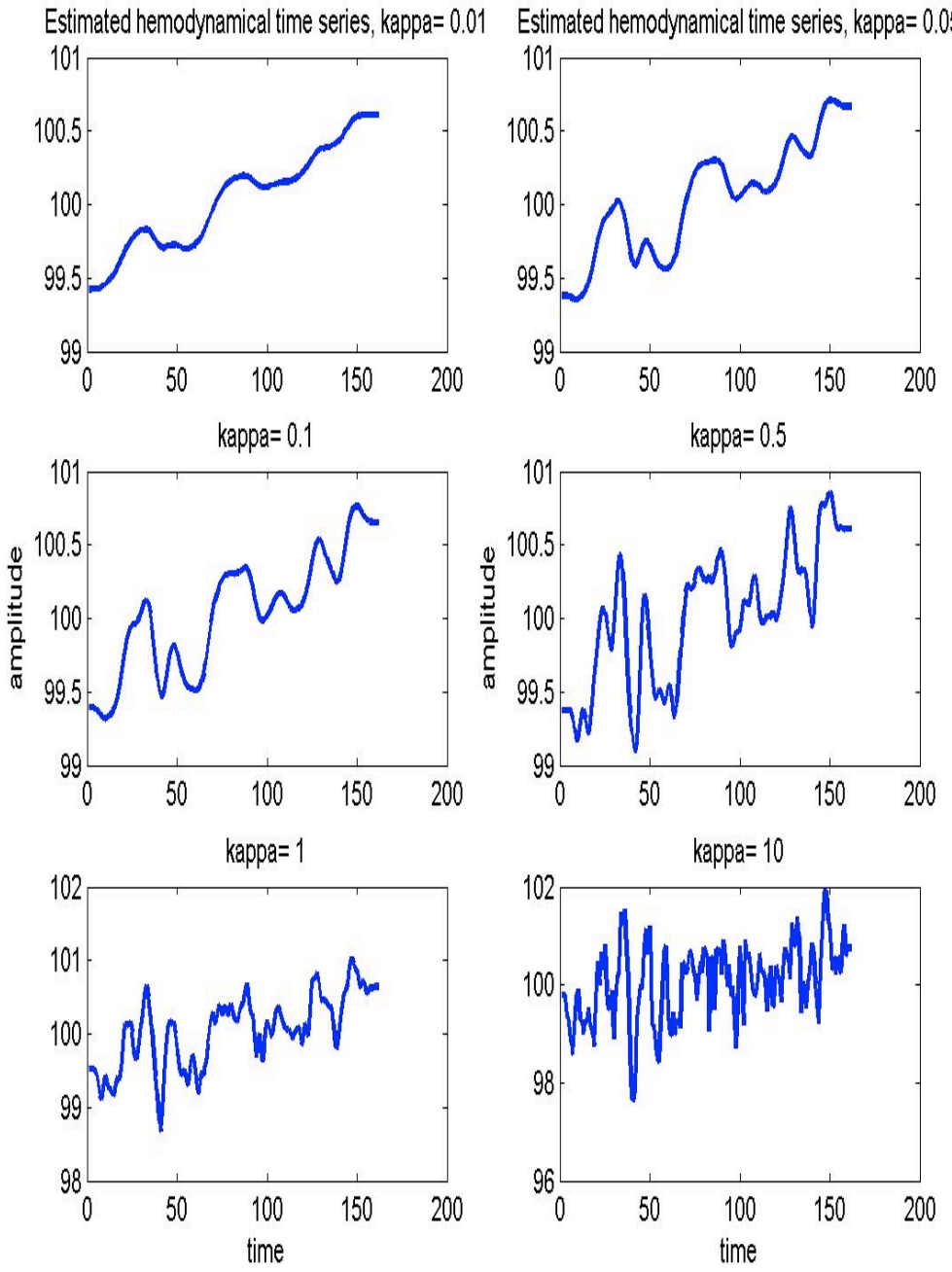


Figure 3.5 The effect of the parameter κ on the algorithm when p=10

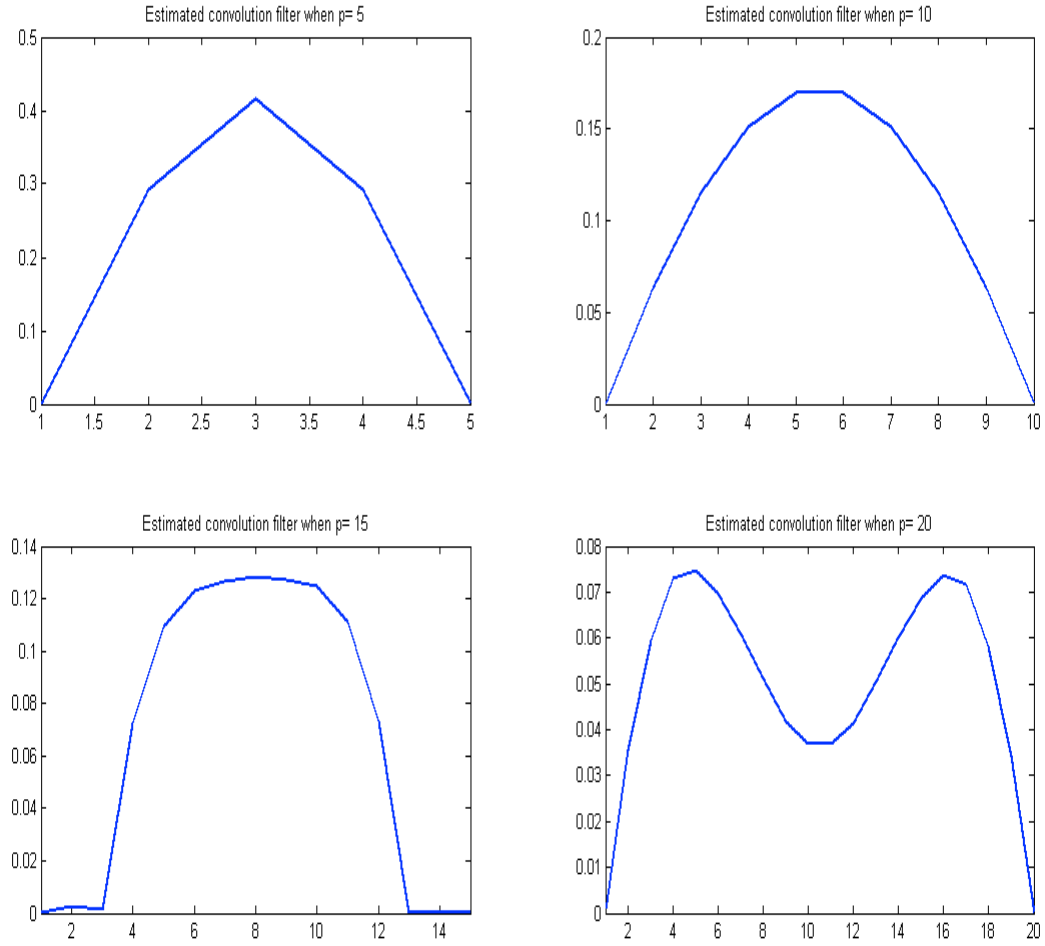


Figure 3.6 The effect of the parameter p on the convolution filter when κ=0.1

The parameter p controls the length of the finite impulse convolution filter. If it is set to 1, then basically it becomes a single impulse which would not change the hemodynamics: $d(t) = k(t) \otimes d(t)$ (up to scaling of the delay). In this case, blind deconvolution turns into denoising of the fMRI. When we increase p, it starts converging to the underlying stimulus of the conducted neural task. For instance, when it is 5 as shown in the upper left of Figure 3.6, the first block of the stimulus is estimated. When it is 10, it still estimates the first block since it has duration more than 5 and even 10. And finally, when it is set to 20, our algorithm captures one block of the underlying stimulus. Note that for these fMRIs, we have categorical stimulus, and each block

stimulus basically consists of two different types of stimulus. This is clearly reflected in our estimated convolution filter when p is 20 where, the two peaks next to each other create a valley shape. So Figure 3.6 illustrates the evolution of the estimated convolution filter as p increases and shows so how it approaches the real stimulus pattern

Figure 3.7 illustrates the corresponding estimated hemodynamics as *p* changes from 5 to 20:

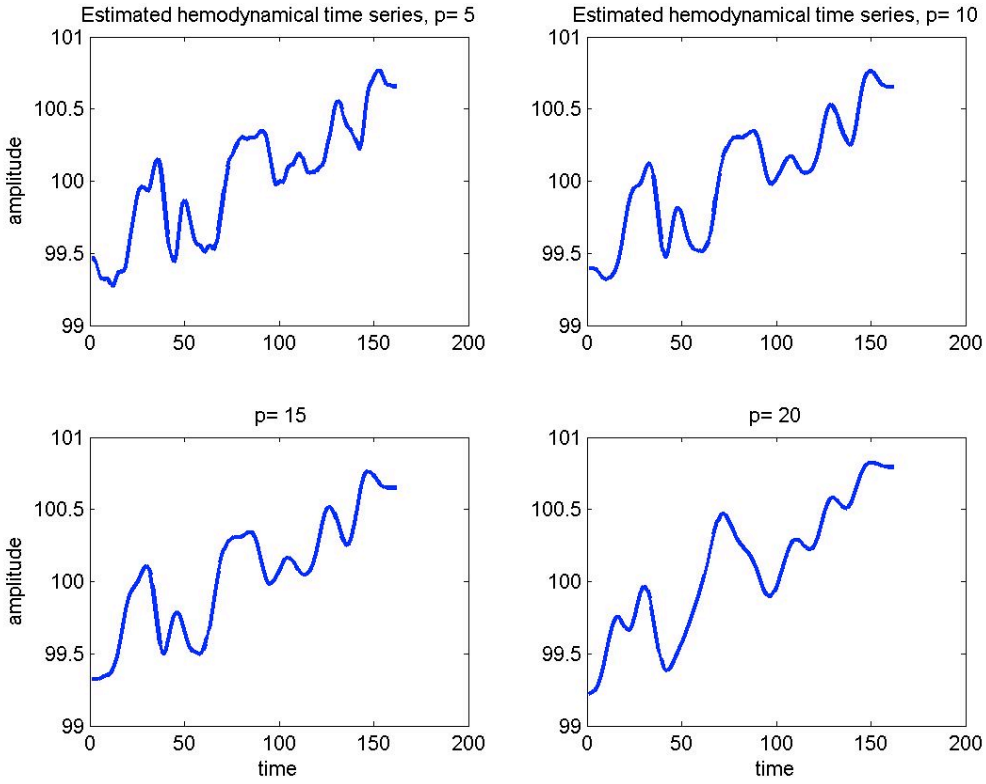


Figure 3.7 The effect of the parameter *p* on the algorithm when κ=10

According to the estimated hemodynamical time series signals, when *p=20* the degree of freedom in the model increases which generates more space for the hemodynamic to be smooth while having not much changes in the data fitting accuracy. As it is already clear from the signals illustrated in Figure 3.7, as *p* increases, they become smoother, meaning that the square sum of the derivatives decreases even though κ is kept fixed. Note that

the smoothest one is the one having p = 20. Comparing the estimated hemodynamical time series of p =5 and p = 10, we notice that they are similar to each other in terms of representing the duration of stimulus blocks as well as their locations. Moreover, when p = 20, the estimated convolution filter extends to the second block of the stimulus which completely fades away the peak around the time instant 50 and also the same affect is also observable to a less degree for the ones coming after the time point 100.

In terms of the activation detection between voxels, the most important thing is the neural activity created on a voxel under a neural task. For instance, when a subject is shown a picture, this immediately creates a electro-chemical (neural) activity on the voxels in the visual vortex of the brain. On the other hand, in the regions of the brain which are irrelevant to visual inputs, voxels are insensitive, i.e, no particular neural activity is expected. This neural activity is named as the stimulus reaching the voxels estimated as the convolution filter signal in our blind deconvolution. For this reason, and in terms of the activation detection through clustering, one should use either the estimated stimulus or the hemodynamical time series which embodies the effect of stimulus. Passive voxels are the ones that the stimulus is not reaching, however they are still subject to some kind of stimulus due to the resting state neural activities. Actually, these resting state neural activities are always existent in the brain which can be modeled as noise, a stochastic delta process. Due to this noise, we think that instead of using the estimated stimulus directly, clustering the hemodynamical time series for selecting the activity type of the brain voxels is more reasonable. If one can capture one block of a stimulus pattern as the convolution filter in our model, then the other component *d(t)* will basically have a nice smooth peak located on the time points of each stimulus block for active voxels, named hemodynamical time series in this thesis. We believe, that is the ideal input for clustering. In this chapter, we analyze our blind deconvolution in terms of the hemodynamical time series estimation. For a good estimation we try to show that the length of the convolution filter should not be big such that it extends to the second block of the stimulus. Optimally, it should just contain a single block because

then the estimated hemodynamic time series includes only the HRFs for every block of stimulus and nothing else. In our cases, that corresponds to p is somewhere between 10 and 20 in value (true value is 18). The effect of p on clustering is also explored in the sensitivity analysis in Chapter 5.

As for HRF extraction, one should use a full length convolution filter meaning that p should be equal or close to the length of the input fMRI signal. Hopefully, then all the blocks of stimulus can be found as *k(t),* then *d(t)* would correctly estimate the HRF. We treat this as a separated issue independent of the clustering. So we analyze this in the HRF extraction part in Section 4.1.

3.2 Preprocessing

Often fMRI signals are known to suffer from low SNR due to several subject or hardware dependent conditions. Our MAP based blind deconvolution algorithm increases the SNR for every fMRI signal, extracting basically the signal triggered only by the neural activity. In our real data experiments, we use the parameters for our blind deconvolution as $\kappa = 0.1 or 1, p = 10$ in order to obtain the input signals for clustering. These parameters are going to be discussed in a more detailed way in the sensitivity analysis in Chapter 5.

After we extract the hemodynamics, we realize a significant drift effect on many of the signals although our raw fMRI data are said to be preprocessed. So we propose a simple preprocessing algorithm to eliminate the so-called 'drifts' from the hemodynamics, that is to say, given the hemodynamics, we then fit a quadratic function to estimate the overall trend in the signal and then simply subtract it from the signal. It works in the linear case as follows:

Given an hemodynamical signal (d_i), we want to approximate it with $d_i^{'} = a^* i + b^*$, so

$$\begin{aligned}(a^*, b^*) &= \arg\min_{(a,b)}\left(\sum_i (d_i - ai - b)^2\right) \\ &= \arg\min_{(x)} (Ax - d)^T (Ax - d)\end{aligned} \tag{3.41}$$

where

$$A = \begin{bmatrix} 1 & 1 \\ \vdots & \vdots \\ i & 1 \\ \vdots & \vdots \\ N & 1 \end{bmatrix}, \qquad x = \begin{bmatrix} a \\ b \end{bmatrix}, \qquad d = \begin{bmatrix} d_1 \\ \vdots \\ d_i \\ \vdots \\ d_N \end{bmatrix}$$

And the solution is given as:

$$x = (A^T A)^{-1} A^T d \tag{3.42}$$

$$d\,' = Ax \tag{3.43}$$

As for the case of quadratic fitting, we update the matrices above such that they also incorporate the square terms:

$$A = \begin{bmatrix} 1 & 1 & 1 \\ \vdots & \vdots & \vdots \\ i^2 & i & 1 \\ \vdots & \vdots & \vdots \\ N^2 & N & 1 \end{bmatrix}, \qquad x = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \qquad d = \begin{bmatrix} d_1 \\ \vdots \\ d_i \\ \vdots \\ d_N \end{bmatrix}$$

Then the solution stays the same as in (3.43) and d_i is estimated to be $d_i = ai^2+bi+c$.

We standardize each filtered time course by subtracting the mean and dividing it by its standard deviation. In the absence of standardization, cross-correlation between two signals could be dominated by similar signal variances rather than similar patterns [56], [75].

Algorithm of preprocessing:

Input: Raw fMRI, *r*

1. *d* ← extract_hemodynamic(*r*)
2. Find *d'* by quadratic fitting
3. *d* ← (*d*-mean(*d*)) / std (*d*)
4. Return *D*=*d*-*d'*

Figure 3.8 illustrates the pre-processing on a real fMRI example:

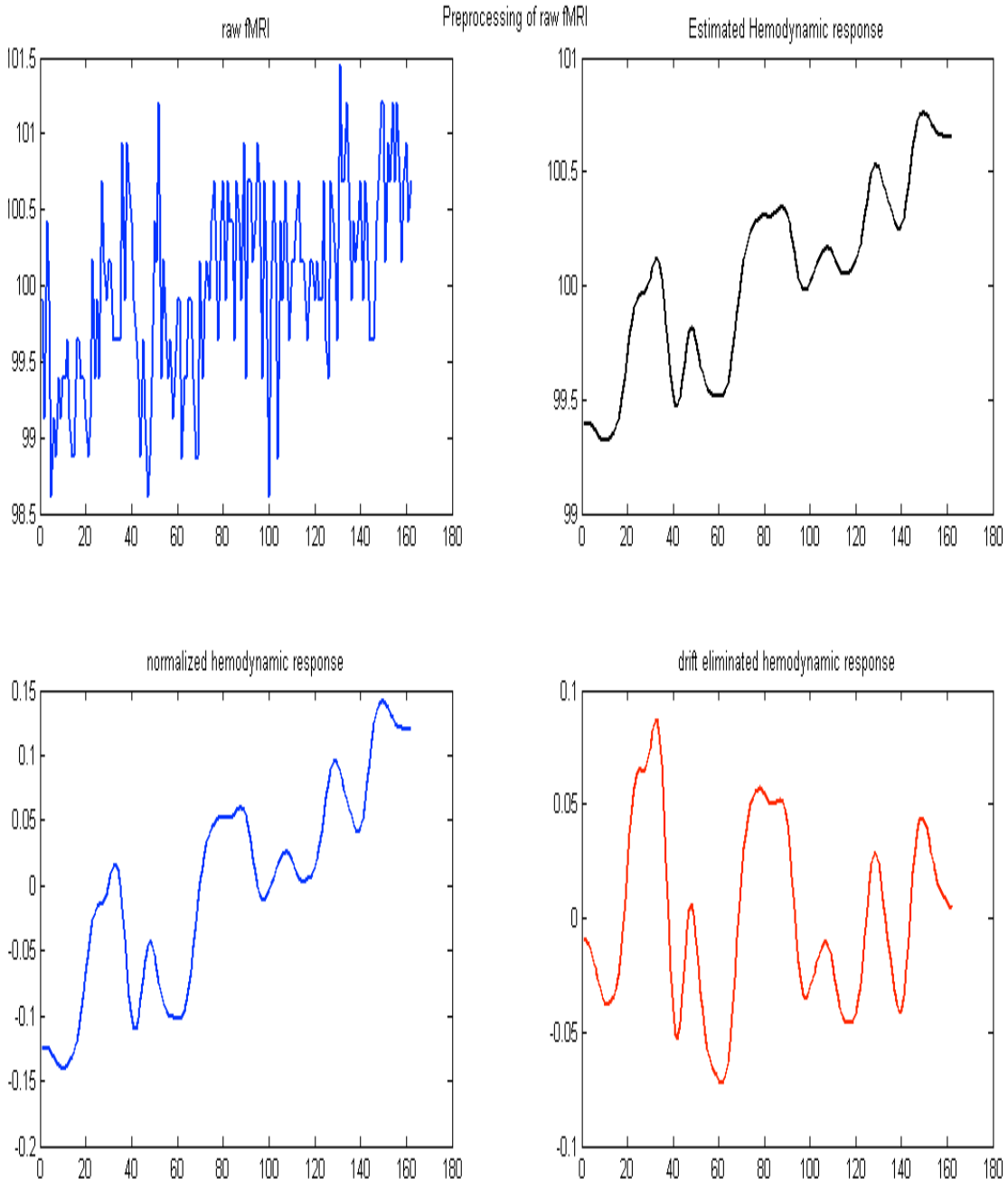


Figure 3.8 Obtaining drift eliminated hemodynamic response

After we preprocess “hemodynamical time series” which is the output of our MAP Blind deconvolution algorithm, we use them as input to the clustering. In all clustering algorithms, the chosen distance/similarity metric is crucial for the success of the clustering algorithm so this is also valid for spectral clustering that we applied in this thesis. For this reason we compare possible metrics and this comparison showed us that the best results are obtained with Hausdorff distance. Through the following section we will give these comparisons with examples and also show how it fits well to the fMRI problem at hand.

3.3 Hausdorff Distance of fMRI

Activation detection is an important topic in fMRI data analysis. Given the fMRI signal for a voxel, a technique is required to declare whether that voxel is active or inactive under a neural task. In this thesis, one of our goals is to cluster the estimated hemodynamics based on their features highly correlated with voxel activation. As we state in the MAP Blind Deconvolution section, peak magnitudes and the locations of the peaks in a given fMRI signal are good features since they are strongly correlated with the stimulus reaching voxels which plays a central on activation. In this respect, we analyze the Hausdorff metric for clustering since we think of it as having powerful properties for activation detection of brain voxels.

Unlike most distance measures, Hausdorff distance measures the distance or similarity between sets. That is, in the context of Hausdorff distance, vectors become sets and ordered points become unordered elements. Accordingly, given a hemodynamical time series which was extracted via MAP Blind Deconvolution D, we map it to a set S via:

$$D =< d_1, d_2, ..., d_n > \rightarrow S = \{s_i : s_i =< d_i, \tau * i >\} \tag{3.44}$$

where τ is the scaling factor in order for the time index *i* not to dominate the hemodynamic magnitudes, d_i. This mapping is due to the definition of the Hausdorff distance and the way it is calculated.

First, we note that Hausdorff distance between S_1 and S_2 is the maximum of the directed Hausdorff distance from S_1 to S_2 and the one from S_2 to S_1 i.e,

$$H(S_1,S_2) = H(S_2,S_1) = \max(h(S_1,S_2), h(S_2,S_1)) \tag{3.45}$$

This is the usual way of generating a 'metric' which is symmetric based on the directed Hausdorff.
Following is the algorithm for its calculation:

Input:

Two time series: $D_1 =< d_{11}, d_{12,}..., d_{1n} >$, $D_2 =< d_{21}, d_{22,}..., d_{2n} >$

Scaling parameter τ

Algorithm:

(1) Switch to set representations:

a. $D_1 =< d_{11}, d_{12,}..., d_{1n} > \rightarrow S_1 = \{s_{1i} : s_{1i} =< d_{1i}, \tau * i >\}$

b. $D_2 =< d_{21}, d_{22,}..., d_{2n} > \rightarrow S_2 = \{s_{2i} : s_{2i} =< d_{2i}, \tau * i >\}$

(2) Calculate the directed Hausdorff $h(S_1,S_2)$ via:

a. For every $s_{1i} \in S_1$,

i. Find $s_{2j} \in S_2$ such the Euclidean distance $dE(s_{1i},s_{2j}) = \sqrt{(d_{1i} - d_{2j})^2 + (\tau(i-j))^2}$ is minimized. So, s_{2j} is the closest element in

set S_2 to the element s_{1i}, and the distance $dE(s_{1i}, s_{2j})$ is the corresponding minimum distance, let us denote it by dE_{1i}.

b. Then, $h(S_1, S_2) = \max(dE_{11}, dE_{12,}, ..., dE_{1n})$

(3) $H(S_1, S_2) = H(S_2, S_1) = \max(h(S_1, S_2), h(S_2, S_1))$

By shifting from vector representations to set representations, we are basically discarding the correspondences that are given by the order/position (or even dimension) of vector components in the vector representation, that is to say, d_{1i} corresponds to d_{2i}. However, the same is not true for s_{1i} and s_{2i}, i.e, we do not say, s_{1i} corresponds to s_{2i} since they are set elements but not vector components.

For the sake of a simple comparison, standard Euclidean distance is a function of differences of the ordered vector components: $dE(D_1, D_2) = \sqrt{\sum (d_{1i} - d_{2i})^2}$. On the other hand, Hausdorff distance discards the correspondences between d_{1i} and d_{2i}. Instead, it constructs its own directed correspondences by finding the closest (closest in a sense with respect to the underlying metric which is usually the Euclidean metric) element $s_{2j} \in S_2$ to a given element $s_{1i} \in S_1$. Hence, with this new form of correspondence schema, d_{1i} does not correspond to d_{2i} but corresponds d_{2j}. The Hausdorff correspondences are directly related to the map from the vector representation to the set representation which should be designed according to the application in hand. For instance one can use the map in (3.46):

$$D =< d_1, d_2, ..., d_n > \rightarrow S = \{s_i : s_i = d_i, i = 1 : n\} \qquad (3.46)$$

In this case Hausdorff correspondences are completely based on the magnitude d_i's, and so the vector positions (index i) are completely discarded.

There can be many choices for this map and it is really a design issue depending on the application. Considering our clustering problem, we choose to use the map:

$$D =< d_1, d_2, ..., d_n > \rightarrow S = \{s_i : s_i =< d_i, \tau * i >\} \tag{3.47}$$

This is a flexible choice for the reason that it provides a nice trade-off between the two extreme via the parameter τ. This is going to be illustrated later in this chapter (pages 115-118) but before that let us continue with a mathematical treatment of our thoughts:

- As $\tau \rightarrow 0$, then our map converges to the map in (3.46):

$$D =< d_1, d_2, ..., d_n > \rightarrow S = \{s_i : s_i = d_i, i = 1 : n\} \tag{3.48}$$

And then Hausdorff distance will completely ignore the positions of vector components, i's, and the correspondences will be completely based on the magnitudes of vector components, d_i's. In this case, for instance, there will be a perfect correspondence from d_5 to d_{500} if $d_5 = d_{500}$ even though d_5 and d_{500} are realized at positions which are very far from each other.

- As $\tau \rightarrow \infty$, the scenario is totally different and the vector positions start dominating the calculation of pairwise distances:

given instances, $s_{1i} \in S_1$ and $s_{2j} \in S_2$, then the distances between them,

$$\sqrt{(s_{1i} - s_{2j})^2} = \sqrt{(d_{1i} - d_{2j})^2 + \tau^2 (i - j)^2} \rightarrow \infty, \qquad if\ i \neq j \tag{3.49}$$

$$\sqrt{(s_{1i}-s_{2j})^2}=\sqrt{(d_{1i}-d_{2j})^2+\tau^2(i-j)^2}\to d_{1i}-d_{2j},\qquad if\ fi=j \tag{3.50}$$

Then, the Hausdorff correspondence will always be from $s_{1i}\in S_1$ and $s_{2i}\in S_2$ i.e, from $s_{1i}\in S_1$ to $s_{2j}\in S_2$ such that *i=j*. Hence, the correspondences will be completely based on the positions of vector components, *i*'s. In this case, directed Hausdorff distances become symmetric since $s_{1i}\in S_1$ corresponds to $s_{2j}\in S_2$ and $s_{2j}\in S_2$ corresponds to $s_{1i}\in S_1$ $H(S_1,S_2)=H(S_2,S_1)=h(S_1,S_2)=h(S_2,S_1)$. And interestingly, the Hausdorff distance converges to L_∞ distance between *D1* and *D2*. The following theorem formalizes our thoughts for the case $\tau\to\infty$.

Theorem: Given two finite length vectors (or time series),

$D_1=<d_{11},d_{12,},...,d_{1n}>$

$D_2=<d_{21},d_{22,},...,d_{2n}>$

and given the maps

$D_1=<d_{11},d_{12,},...,d_{1n}>\to S_1=\{s_{1i}:s_{1i}=<d_{1i},\tau*i>\}$

$D_2=<d_{21},d_{22,},...,d_{2n}>\to S_2=\{s_{2i}:s_{2i}=<d_{2i},\tau*i>\}$

the Hausdorff distance (as defined above) $H(S_1,S_2;\tau)$ converges to L_∞ distance between *D1* and *D2*, $\|D_1-D_2\|_\infty$ as $\tau\to\infty$.

Proof:

For an element $s_{1i}\in S_1$, we say $s_{1i}\in S_1$ corresponds to $s_{2j}\in S_2$ iff

$j=\arg\min_{j\in\{1,...,n\}}\left\{\sqrt{(d_{1i}-d_{2j})^2+\tau^2(i-j)^2}\right\}$.

Then put the sets:

$T_1 = \{(s_{1i}, s_{2j}) : s_{1i} \in S_1 \text{ correspond s to } s_{2j} \in S_2, \text{and } i \neq j\}$

$T_2 = \{(s_{2i}, s_{1j}) : s_{2i} \in S_2 \text{ correspond s to } s_{1j} \in S_1, \text{and } i \neq j\}$.

Claim 1 $\forall \tau$ such that $T_1 \cup T_2 = \phi$,

Then so for every $i \in \{1,2,...,n\}$,

$s_{1i} \in S_1$ corresponds to $s_{2i} \in S_2$ and $s_{2i} \in S_2$ correspond s to $s_{1i} \in S_1$.

$$\begin{aligned} h(S_1, S_2; \tau) &= \max_{i \in \{1,2,...,n\}} \left\{ \sqrt{(d_{1i} - d_{2i})^2 + \tau^2 (i-i)^2} \right\} \\ &= \max_{i \in \{1,2,...,n\}} \left\{ \sqrt{(d_{1i} - d_{2i})^2} \right\} \\ &= \left\| D_1 - D_2 \right\|_\infty \end{aligned} \tag{3.51}$$

Because of the symmetry,

$$\begin{aligned} h(S_2, S_1; \tau) &= \max_{i \in \{1,2,...,n\}} \left\{ \sqrt{(d_{2i} - d_{1i})^2 + \tau^2 (i-i)^2} \right\} \\ &= \max_{i \in \{1,2,...,n\}} \left\{ \sqrt{(d_{2i} - d_{1i})^2} \right\} \\ &= \left\| D_2 - D_1 \right\|_\infty \end{aligned} \tag{3.52}$$

Then obviously,

$$\begin{aligned} H(S_1, S_2; \tau) &= h(S_1, S_2; \tau) = h(S_2, S_1; \tau) \\ &= \left\| D_1 - D_2 \right\|_\infty = \left\| D_2 - D_1 \right\|_\infty \end{aligned} \tag{3.53}$$

Claim 2 If for some τ^*, $T_1 \cup T_2 \neq \phi$, then there exists $p \in \Re$ and $p > \tau^*$ such that $\forall \tau \geq p$, $T_1 \cup T_2 = \phi$.

Pick an element $(s_{1i}, s_{2j}) \in T_1$.

Then since $s_{1i} \in S_1$ corresponds to $s_{2j} \in S_2$ and $i \neq j$,

$|d_{1i} - d_{2i}| > \sqrt{(d_{1i} - d_{2j})^2 + \tau^2 (i-j)^2}$; where $\tau = \tau^*$.

However, $\forall \tau \geq \sqrt{\frac{(d_{1i}-d_{2i})^2}{(i-j)^2}}$,

$$\sqrt{(d_{1i}-d_{2j})^2+\frac{(d_{1i}-d_{2i})^2}{(i-j)^2}(i-j)^2}=\sqrt{(d_{1i}-d_{2j})^2+(d_{1i}-d_{2i})^2}>\left|d_{1i}-d_{2i}\right| \Rightarrow$$

$s_{1i} \in S_1$ corresponds to $s_{2i} \in S_2$.

Then put $p_{1i}=\sqrt{\frac{(d_{1i}-d_{2i})^2}{(i-j)^2}}$, $p_1=\max_{\forall (s_{1i},s_{2j})\in T_1}\{p_{1i}\}$, so,

$\forall \tau \geq p_1$,

$\forall i, p_1 \geq p_{1i} \Rightarrow s_{1i} \in S_1$ correspond s to $s_{2i} \in S_2$

and we can similarly define p_{2i}, p_2, also put $p=\max\{p_1, p_2\}$.

Hence, obviously, $\forall \tau \geq p$,

$s_{1i} \in S_1$ correspond s to $s_{2i} \in S_2$ and $s_{2i} \in S_2$ correspond s to $s_{1i} \in S_1$.

$\Rightarrow T_1 \cup T_2 = \phi$ (by definition)

Finally, by claim 1, since $\forall \tau \geq p$, $T_1 \cup T_2 = \phi$,

$$H(S_1,S_2;\tau \geq p)=h(S_1,S_2;\tau \geq p)=h(S_2,S_1;\tau \geq p)$$
$$=\left\|D_1-D_2\right\|_\infty=\left\|D_2-D_1\right\|_\infty$$

This completes the proof.

As a result, as the parameter τ changes in the interval $[0,\infty]$, we obtain a nice trade-off between having correspondences of the vector components based on alone their positions versus having those correspondences based on their magnitudes alone. These are the two limits of the trade-off. When $\tau = \infty$, we obtain hard correspondences, which means that the magnitude values are completely ignored. On the other hand, when $\tau = 0$, this time the positions are ignored and the correspondences are based on magnitudes. And any value within this interval provides the trade-off.

By fine-tuning this parameter, the Hausdorff distance gains the capability of establishing soft correspondences between the time instants of a time series and the neighboring time instants of another one. For example in the above example, d_{1i} can possibly correspond to d_{2j} if *i* is in the neighborhood of *j*. And the parameter τ precisely controls the size of this neighborhood. If it is zero, the size of this neighborhood is also zero and so d_{1i} is forced to correspond to d_{2i}. If we tune it, we can adjust the size of this neighborhood and d_{1i} can correspond to d_{2j} such that they are close to each other in time and $\left|d_{1i} - d_{2j}\right|$ is sufficiently smaller than $\left|d_{1i} - d_{2i}\right|$. This is one of the nice properties of Hausdorff distance especially for the hemodynamic clustering problem because it often happens that the hemodynamics from a voxel turns out to be a delayed version of the hemodynamics of another voxel. In such cases, one wants to take care of such delays instead of directly assigning a large distance to them due to incorrect correspondences of the time instants. Let us consider a hypothetical example:

Let $D_1 =< d_{11}, d_{12}, ..., d_{1n} > \rightarrow S_1 = \{s_{1i} : s_{1i} =< d_{1i}, \tau * i >\}$; $d_{1i} = \mathrm{A} * \sin(\mathrm{wi})$,

$D_2 =< d_{21}, d_{22}, ..., d_{2n} > \rightarrow S_2 = \{s_{2i} : s_{2i} =< d_{2i}, \tau * i >\}$; $d_{2i} = \mathrm{B} * \sin(\mathrm{wi} - \phi)$, and

$A = 10,\ B = 15,\ w = 0.0125,\ \phi = 1.25,\ \tau = 0.01,\ n = 1000.$

So we have two sinusoids, one D_2, has larger peak-to-peak amplitude and a phase shift with respect to D_1. We show the aforementioned property of Hausdorff distance on these

signals. Ideally one can argue that a good distance metric between D_1 and D_2 should be a function of $(A\text{-}B)$ and φ which may be $dis\tan ce(D_1,D_2) = \sqrt[p]{(A\text{-}B)^p + c^p \varphi^p}$ where c is a weighting factor for the phase difference. For instance, the Euclidean distance $dE(D_1, D_2) = \sqrt[2]{(d_{1i} - d_{2i})^2}$, then would not be a good choice because it uses hard correspondences and never considers possible phase differences or shifts between its arguments. On the other hand, we expect the Hausdorff distance to be a better choice and closer to reality due to the aforementioned and discussed property. Figure 3.9 illustrates these signals:

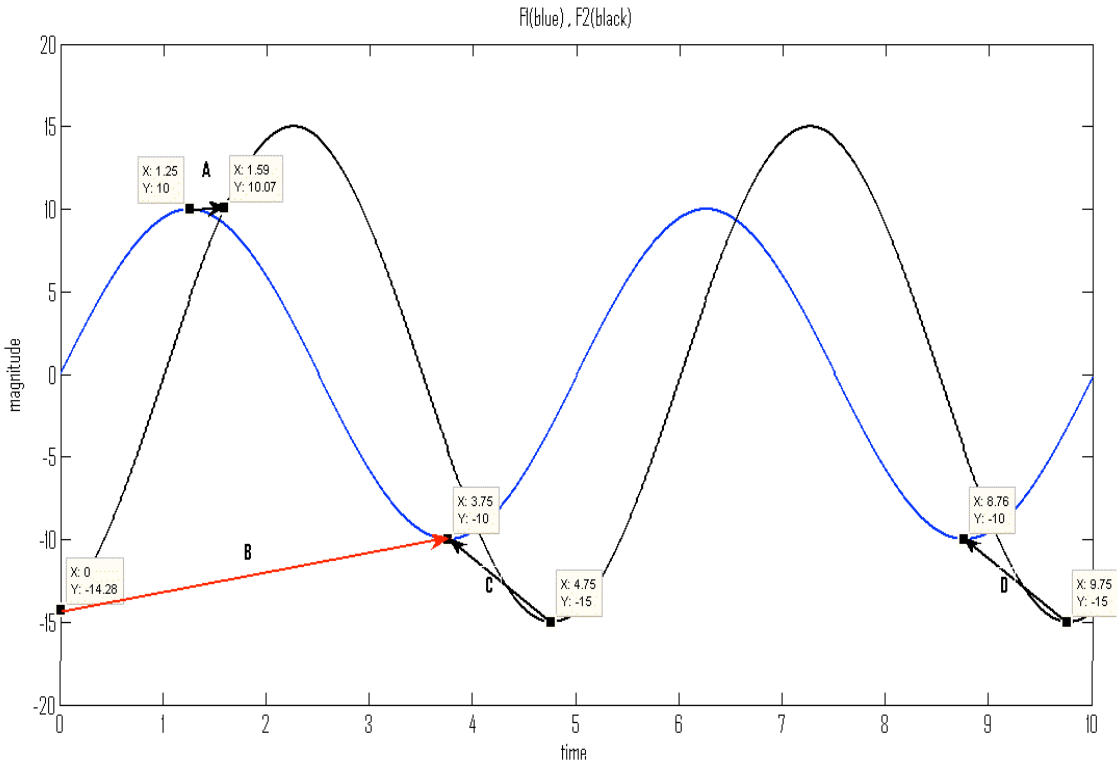


Figure 3.9 Hausdorff distance between two sinusoidal signals

D_1 is the blue colored sinusoid and D_2 is black and as illustrated D_2 is 100 seconds (time axis is shown as scaled by the parameter $\tau = 0.01$) shifted rightward. The distances calculated by the Euclidean and Hausdorff are as follows:

Euclidean = 341.6783

Hausdorff = 5.6759 (which is precisely the length of red arrow)

First thing to note is that the Euclidean distance generates an unnecessarily large distance due to the misalignment of the two signals. On the other hand Hausdorff distance generates a reasonable distance or at least satisfies our expectations since it is only related to the real deviation between the two signals in amplitude and phase. The arrows are the detected correspondences among which, the ones from D_1 to D_2 are not good matches because Hausdorff distance catches the similar or even same magnitude values from D_1 to D_2. This is mainly because the range of D_2 covers the range of D_1 and so it becomes trivial to find a similar magnitude value in D_2 with respect to the magnitude of a given point in D_1 since the distance calculation in this example is not very sensitive to the time due to a relatively low value of $\tau = 0.01$. The arrow A is an example of such incorrect correspondences. Note that its tail is on D_1 and its head is on D_2 which means the correspondence is from D_1 to D_2. As it is expected, the point (125, 10) in D_1 corresponds to (159, 10.07) in D_2. Tail of this match is the peak magnitude of D_1 and on the other hand, the head of this match is only 66% of the amplitude of D_2 , meaning that a peak is matched to a hill. Similarly, all correspondences from D_1 to D_2 are not good matches. However, Hausdorff distance is wisely not affected from such incorrect matches because at the final step of its algorithm, a maximum is taken between the two directed Hausdorff distances, that is to say, that Hausdorff distance is defined to be the maximum of length of all such correspondences, both from D_1 to D_2 and D_2 to D_1. As for the correspondences from D_2 to D_1, they are all correct correspondences, and so the two signals are clearly aligned well with the help of these correspondences. However, an interesting thing is happening that the red arrow (arrow B) is matching to far minima (still both minimums are in the same cycle of the sinusoids) but the others

(arrows C and D) are matching to close minima. To understand this, one should note that the second sinusoid D_2 is 100 seconds ahead of the first sinusoid, and so the correspondences C and D are pointing leftward in time, which are correct. However, the minimum in the signal D_1 corresponding to the tail of arrow B is not available, and so Hausdorff distance finds the next minimum in the signal D_1, which is approximately 3.75 seconds ahead. This is still a correct match in terms of the peak matches but just the time shift is found to be in the signal D_2 $2\pi - \phi$ but not ϕ. This happens only once and only in the first and last cycles of the signals (in the first cycle, from D_2 to D_1 and in the last cycle from D_1 to D_2). However, one can easily correct this issue by looking at not the maximum of lengths of all correspondences but, for instance, *n'th* maximum (note that the majority of matches are correct, in this example it is 2 correct and 1 incorrect matches) and actually we use this kind of adjustment and we call it 'modified Hausdorff distance' which was explained in Chapter 2.

As a final note on this, after scaling the time with ($\tau = 0.01$), time axis lies within the range [0, 10] and the magnitude range is [-15, 15]. These ranges are comparable and so with the help of time scaling, we obtain the aforementioned trade-off. However, as we state in the theorem, if τ is set to a large value, say 10 in this example, then the range for time becomes [0 10000] and dominates the magnitude values which results in hard correspondences and make the Hausdorff distance become L_∞ distance. Figure 3.10 shows the same example with the scaling $\tau = 10$.

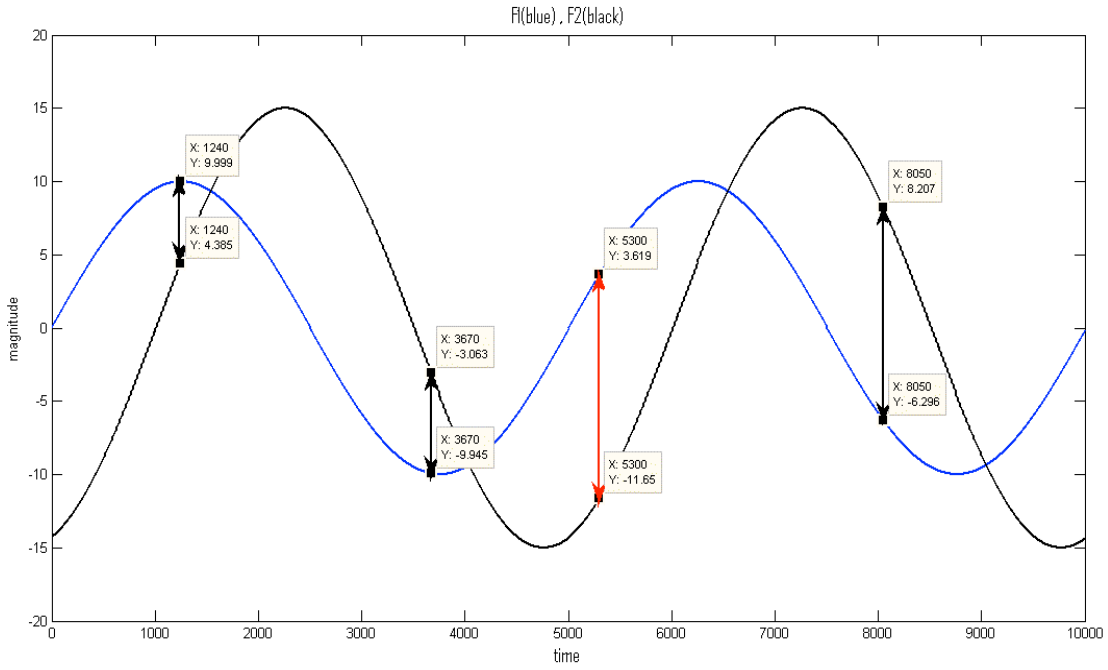


Figure 3.10 The effect of the time scaling parameter, τ, on the Hausdorff distance

In this case the correspondences are shown with double headed arrows since with $\tau = 10$ the directed Hausdorff distances become symmetric, that is to say, s_{1i} corresponds to s_{2i} and s_{2i} corresponds to s_{1i}. This is also clear from the fact that the arrows become vertical. The following are the calculated distances:

Euclidean = 341.6783
Hausdorff = 15.2727
LinfEuclidean = 15.2727

The effect is that the Hausdorff distance in this case becomes larger and far from optimality since the sinusoids here differ from each other not only through amplitude but also in phase and ideally we expect a good metric to be related to both of these parameters. With hard correspondences as in this example, and noting that the Hausdorff distance increased with respect to the previous example two times (5.6759-15.2727), it

becomes too sensitive to the phase shift since it discards the possibility that a phase shift exists. This example is also an illustration of the theorem proved before (page 111).

Next we analyze how Hausdorff distance behaves under some noise conditions. We consider two cases on the same example:

Case 1: We use zero mean unit variance additive independent Gaussian noises added to the signals above at every time instant:

$$D_1 =< d_{11}, d_{12}, ..., d_{1n} > \rightarrow S_1 = \{s_{1i} : s_{1i} =< d_{1i}, \tau * i >\}$$

$$d_{1i} = \mathrm{A} * \sin(\mathrm{wi}) + \sigma * \mathrm{N}(0,1),$$

$$D_2 =< d_{21}, d_{22}, ..., d_{2n} > \rightarrow S_2 = \{s_{2i} : s_{2i} =< d_{2i}, \tau * i >\}$$

$$d_{2i} = \mathrm{B} * \sin(\mathrm{wi} - \phi) + \sigma * \mathrm{N}(0,1), \text{ and}$$

$$A = 10,\ B = 15,\ w = 0.0125,\ \phi = 1.25,\ \tau = 0.01,\ n = 1000,\ \sigma = 1.$$

Figure 3.11 shows an example of a realization of the above sinusoidal signals:

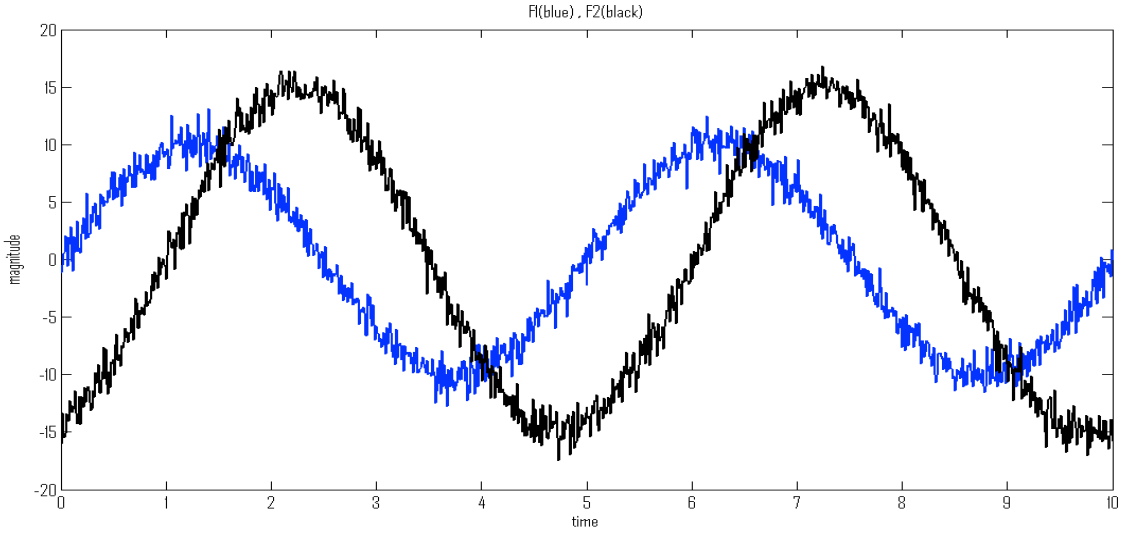


Figure 3.11 Hausdorff distance between two sinusoidal signals under zero mean unit variance additive independent Gaussian noises added to the signals above at every time instant

Case 2: We use zero mean unit variance additive independent Gaussian noises added to the signal amplitudes:

$$D_1 =< d_{11}, d_{12}, ..., d_{1n} >\rightarrow S_1 = \{s_{1i} : s_{1i} =< d_{1i}, \tau * i >\}$$

$$d_{1i} = (\mathrm{A} + \sigma * \mathrm{N}(0,1)) * \sin(\mathrm{wi}),$$

$$D_2 =< d_{21}, d_{22}, ..., d_{2n} >\rightarrow S_2 = \{s_{2i} : s_{2i} =< d_{2i}, \tau * i >\}$$

$$d_{2i} = (\mathrm{B} + \sigma * \mathrm{N}(0,1)) * \sin(\mathrm{wi} - \varphi), \text{ and}$$

$$A = 10,\ B = 15,\ w = 0.0125,\ \phi = 1.25,\ \tau = 0.01,\ n = 1000,\ \sigma = 1.$$

Figure 3.12 shows an example of 100 realizations of the signals above, the blue region contains all the D_1 realizations and the black region contains all the D_2 realizations.

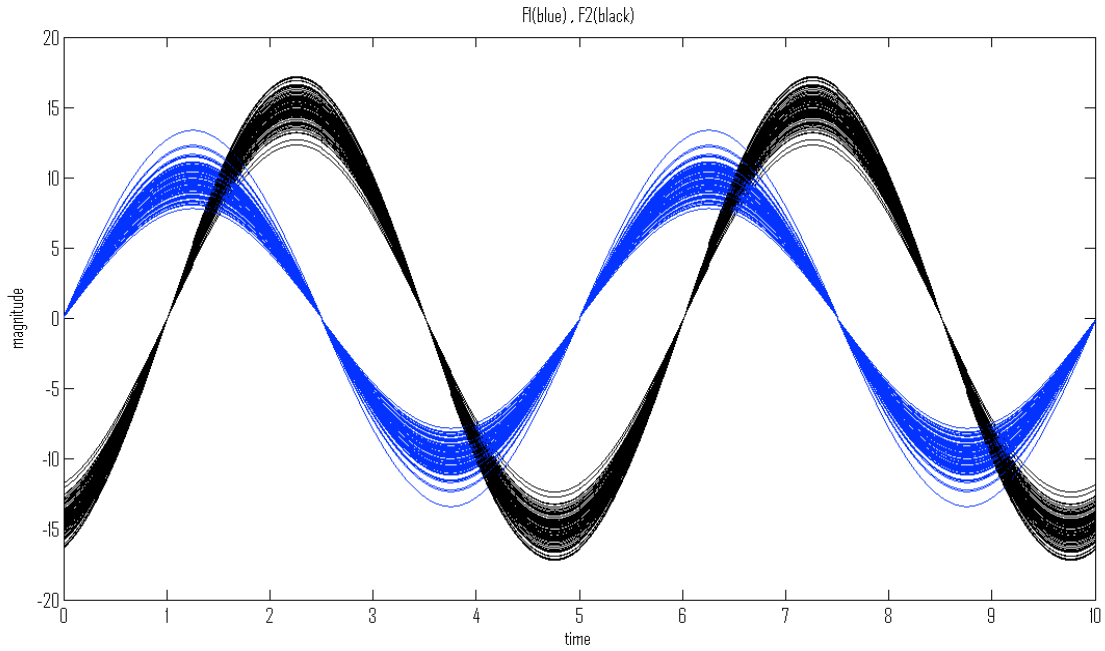


Figure 3.12 Hausdorff distance between two sinusoidal signals under zero mean unit variance additive independent Gaussian noises added to the signal amplitudes

Case 3: We use zero mean unit variance additive independent Gaussian noises added to the phase of signal D_2:

$$D_1 =< d_{11}, d_{12}, ..., d_{1n} > \rightarrow S_1 = \{s_{1i} : s_{1i} =< d_{1i}, \tau * i >\}$$

$$d_{1i} = \mathrm{A} * \sin(\mathrm{wi}),$$

$$D_2 =< d_{21}, d_{22}, ..., d_{2n} > \rightarrow S_2 = \{s_{2i} : s_{2i} =< d_{2i}, \tau * i >\}$$

$$d_{2i} = \mathrm{B} * \sin(\mathrm{wi} - \phi + \sigma * \mathrm{N}(0,1)),$$

and $A = 10,\ B = 15,\ w = 0.0125,\ \phi = 1.25,\ \tau = 0.01,\ n = 1000,\ \sigma = 1.$

Figure 3.13 shows an example of 5 realizations of the signal D_2 and also the D_1 signal.

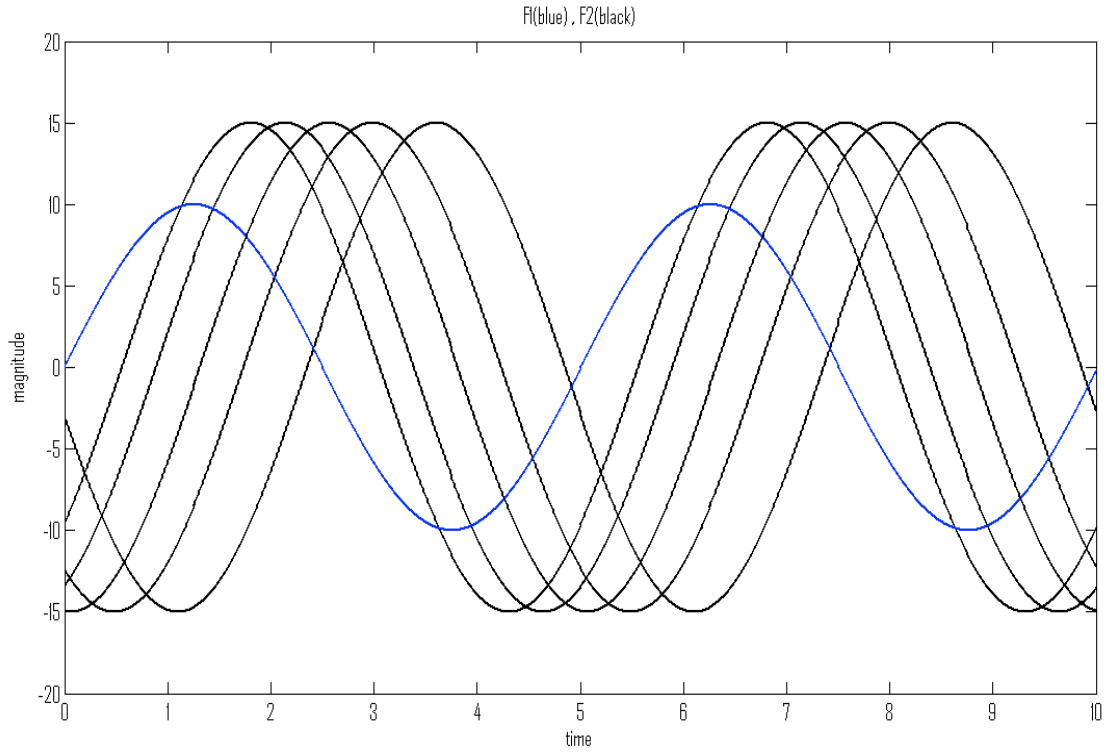


Figure 3.13 Hausdorff distance between two sinusoidal signals under zero mean unit variance additive independent Gaussian noises added to the phase of signal D_2

In order to have a statistically meaningful analysis of the behavior of Hausdorff distance under such noise conditions, we generate such signals 100 times and calculate the Hausdorff distance; Table 3.1 shows the mean and standard deviations of our findings:

	Mean (Hausdorff(D_1, D_2))	Std (Hausdorff(D_1, D_2))
No Noise	5.6759	0
Additive Noise at every time instant (zero mean, unit variance, Gaussian)	5.7463	0.4550
Additive Amplitude noise (zero mean, unit variance, Gaussian)	5.9194	1.0891
Additive Phase noise (zero mean, unit variance, Gaussian)	5.476	0.49

Table 3.1 The mean and standard deviations of HD under different noise conditions

According to our results, the standard deviation is approximately 1 when a unit variance and zero mean Gaussian noise is added to the amplitudes. This shows that Hausdorff distance with the settings above is directly and proportionally responsive to any amplitude changes. One can argue the same for the case when the same noise is added to the phase shift. Although the standard deviation is approximately 0.5 which may seem a little low, the distance measure prove to be still reasonably large since the distance is calculated together with the amplitude, which has a greater effect on the final distance value than the phase shifts due to a small $\tau = 0.01$. And lastly, Hausdorff distance is a little more sensitive than our expectations to the point wise independent Gaussian noises. This is because along a long time interval (in our example 1000), whenever a large noise happens, it directly affects the Hausdorff distance since it is based on the maximum of lengths of all correspondence vectors. However, it is still on an acceptable level since the most important thing is the responsiveness to amplitude and phase shifts which is,

according to our findings, good enough to dominate the point wise Gaussian noise conditions. Furthermore, for the fMRI-clustering problem, we work with smooth hemodynamics, so usually we do not expect point wise noise effect on the hemodynamics.

One important drawback of Hausdorff distance is that it is overly sensitive to outliers, which, in the case of time series data, often happen in a relatively short interval of time. This is really expected because Hausdorff distance in its original form is directly the maximum of the length of aforementioned pointwise correspondences. Hence, whenever we observe a sudden magnitude-wise unusual activity that we call ‘outliers’, in a given time series data, this might result in a severe, wrong correspondence at one time instance at least and then the Hausdorff distance immediately gets affected by this. So it is overly sensitive to such outliers. To illustrate this effect, we simulate this kind of outliers by a Gaussian wavelet as in Figure 3.14:

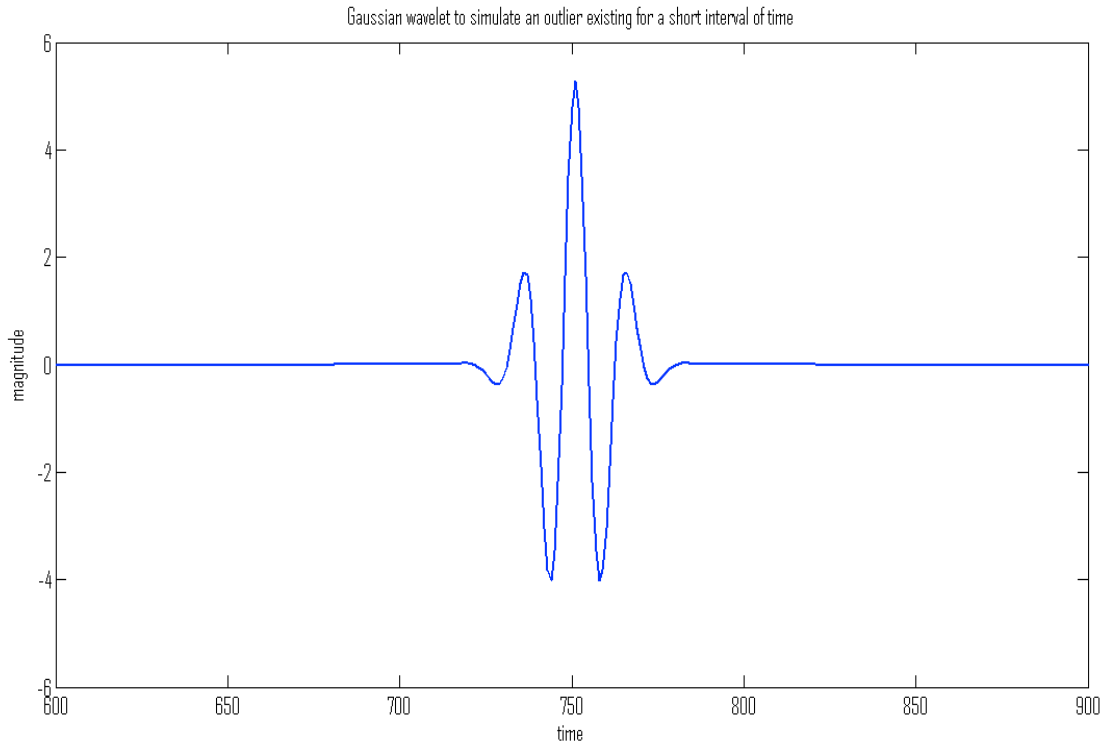


Figure 3.14 Gaussian wavelet to simulate an outlier for a short interval of time

Then we add this Gaussian wavelet representing a possible outlier to the signal D_2 as follows:

Case 4: Additive Gaussian wavelet of short support as outliers

$D_1 =< d_{11}, d_{12}, ..., d_{1n} > \rightarrow S_1 = \{s_{1i} : s_{1i} =< d_{1i}, \tau * i >\}$

$d_{1i} = \mathrm{A} * \sin(\mathrm{wi})$, ($D_2$ stays unchanged.)

Ψ ='Gaussian wavelet of order 8' generated by the MATLAB command: *gauswavf*. It is illustrated in Figure 3.14.

$D_2 =< d_{21}, d_{22}, ..., d_{2n} > \rightarrow S_2 = \{s_{2i} : s_{2i} =< d_{2i}, \tau * i >\}$

$d_{2i} = \mathrm{B} * \sin(\mathrm{wi} - \phi) + \Psi(\mathrm{i})$, and

$A = 10,\ B = 15,\ w = 0.0125,\ \phi = 1.25,\ \tau = 0.01,\ n = 1000,\ \sigma = 1.$

The resulting signals are as in Figure 3.15.

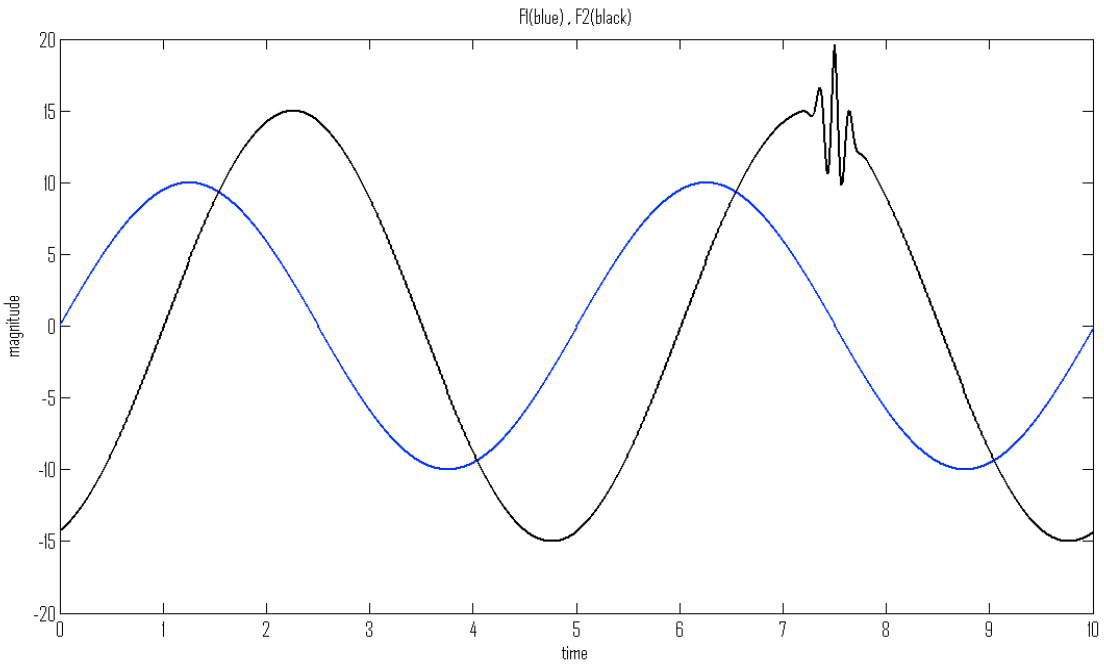


Figure 3.15 Two sinusoidal signals under additive Gaussian wavelet of short support as outliers

Euclidean = 342.044

Hausdorff = 9.6739

In this case the Hausdorff distance turns out to be approximately 9.6739, which is approximately 70% more than the distance value without the outlier: 5.6759. This proves that even such an outlier of a short time interval severely affects the Hausdorff distance. However, we can make it more robust by using the 'modified Hausdorff distance', which was defined in Chapter 2. Recall that in the case of modified Hausdorff distance, instead of calculating the very maximum deviation between two given signals, we calculate the n'th maximum. The benefit of using modified Hausdorff distance is as follows: As long as the time duration of the unusual activity is short enough compared to the total length of the time series, then the majority of the point correspondences should still be correct, in other words, when the outlier is of a short period of time, then the wrong correspondences resulting from this outlier will not be many. Then just by

calculating the, say, n'th maximum of length of all correspondences, Hausdorff distance should then again be producing a reasonable distance, because in this case, it would be not sensitive to the first n-1 correspondences which would be wrong due to the outlier. Figure 3.16 shows the modified Hausdorff distance vs different choices for the 'coverage rate' or the α parameter of the modified Hausdorff distance under different conditions:

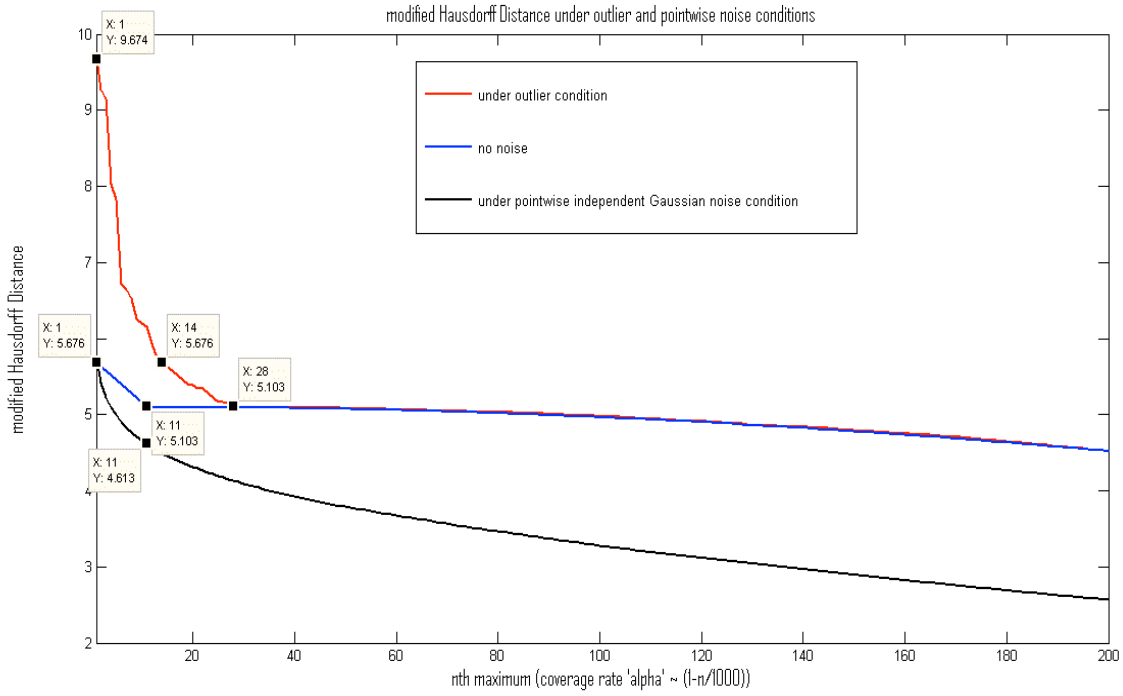


Figure 3.16 Modified Hausdorff distance with different α parameters under outlier, no noise and Gaussian noise conditions.

The 'blue' curve in Figure 3.16 shows the behavior of modified Hausdorff distance between the original signals D_1 to D_2 (no noise and no outliers) versus different coverage rates. So when the coverage is 100% we recover the original Hausdorff distance, which is 5.676 as we reported before. As shown in the figure, there is a sudden drop between the 100% coverage (1^{st} maximum) and 99%coverage (11^{th} maximum). This is due to the fact that even though the signal D_2 is ahead of D_1 by ϕ, some of the

correspondences are found as if the phase difference was $2\pi - \phi$. In other words, because the infinite extension of the signals are not available or because the two signals start at different positions within their cycles, for some of the points in D_2 the correct correspondences do not exist in D_1 which results in some of the points in D_2 being matched in the next cycle in D_1. However, with the help of the modified distance, if we operate at the end of the initial drop, i.e, in ($\alpha = 0.99, 11th$ maximum) we overcome this issue as long as the signals are long enough such that misalignment of the initialization of the signals cannot dominate the overall time series. And so at $\alpha = 0.99, 11th$ maximum, the distance turns out to be 5.103 which is approximates the ideal distance: $\sqrt{(\mathrm{A}-\mathrm{B})^2 + (\varphi)^2} = \sqrt{(10-15)^2 + (1.26)^2} = 5.1539.$

As for the case when there exists an outlier, modified Hausdorff distance still helps to a great degree. The 'red' curve in the above figure shows our findings for this case: If we set ($\alpha = 0.97, 28th$ maximum), then we approximately recover the ideal distance again, 5.103, and so we successfully discard the wrong correspondences due to the outlier.

On the other hand, the situation is a little different when there exists pointwise independent unit variance zero mean Gaussian noise in signals. The 'black' curve shows the average of modified Hausdorff distances over 100 experiments with the same parameters under this noise conditions. As expected, when $\alpha = 1, 1st$ maximum, then the average is equal to the original distance because the effect of zero mean noise is faded away. However, if we set $\alpha = 0.99, 11th$ maximum then there is an offset of 0.5 compared to the no noise case. We think, this is because in an interval of 50 seconds, it is very likely that one of the realizations of 50 Gaussian noise components results in a wrong correspondence, which in turn generates an offset of $50 * \tau(0.01) = 0.5$. Through Tables 3.2, 3.3, 3.4, and 3.5, we summarize our findings on our hypothetical examples:

$\alpha = 1, 1st$ maximum	Mean(Hausdorff(D_1,D_2))	Std (Hausdorff(D_1,D_2))
No Noise	5.6759	0
Additive Noise at every time instant (zero mean, unit variance, Gaussian)	5.7463	0.4550
Additive Amplitude noise (zero mean, unit variance, Gaussian)	5.9194	1.0891
Additive Phase noise (zero mean, unit variance, Gaussian)	5.476	0.49
Outlier (Gaussian wavelet of order 8)	9.6739	0

Table 3.2 The mean and standard deviations of standard HD under different noise conditions

$\alpha = 0.99, 11th$ maximum	Mean (modified Hausdorff(D_1,D_2))	Std (modified Hausdorff(D_1,D_2))
No Noise	5.103	0
Additive Noise at every time instant	4.669	0.2356
Additive Amplitude noise	5.2116	1.0891
Additive Phase noise	5.4332	0.45
Outlier	6.1581	0

Table 3.3 The mean and standard deviations of modified HD with α=0.99 under different noise conditions

$\alpha = 0.97, 30th$ maximum	Mean (modified Hausdorff(D_1,D_2))	Std (modified Hausdorff(D_1,D_2))
No Noise	5.0931	0
Additive Noise at every time instant	4.1104	0.2021
Additive Amplitude noise	5.2175	1.3380
Additive Phase noise	5.3231	0.3400
Outlier	5.0993	0

Table 3.4 The mean and standard deviations of modified HD with α=0.97 under different noise conditions

$\alpha = 0.95, 50th$ maximum	Mean(modified Hausdorff(D_1,D_2))	Std(modified Hausdorff(D_1,D_2))
No Noise	5.0724	0
Additive Noise at every time instant	3.799	0.1819
Additive Amplitude noise	5.451	1.4246
Additive Phase noise	5.1441	0.1915
Outlier	5.0846	0

Table 3.5 The mean and standard deviations of modified HD with α=0.95 under different noise conditions

Noting the mean and standard deviation values in the above tables for the case of additive white Gaussian noise at every time instant, there emerges the variance-bias trade-off, since as the coverage rates drops, the modified Hausdorff distance becomes more and more robust to noise and the standard deviation declines. On the other hand, the average distance value becomes far from the original distance value in the case of no noise which brings a bias. For instance, when using the 50^{th} maximum in the modified Hausdorff distance, we get the lowest standard deviation value 0.18 shown in table 3.5. This low value of standard deviation means that the distance is insensitive to the additive noise. On the other hand and for the same case, the mean value of the distance is 3.79 which is 5.07 – 3.8 = 1.27 offset from the true mean. This is the bias aforementioned.

As we already explained, modified Hausdorff distance is robust to outliers, i.e, unusual activity of a small time interval. Note that until using the 11^{th} maximum as shown in tables 3.3 and 3.2, the modified Hausdorff distance is far from the true mean. However if we use the 30^{th} maximum or more, it turns out to be very close to the true mean in Tables 3.5 and 3.4.

Changes in the amplitudes of the signals directly show their effects in the calculated modified Hausdorff distances in all of the 4 tables above. Note that the standard deviations for this case are large, between 1 and 1.4, which is due to the unit variance amplitude changes we applied. This is reasonable and also desirable, because we want a good metric sensitive to general amplitude changes as well as the phase changes.

As for the phase changes, the modified Hausdorff distance becomes less sensitive as the coverage rate decreases. This is not desirable but we, at least, want a good metric to be more sensitive to the phase changes than it is to the point-wise noise. And this is satisfied according to our findings. Note that, the standard deviation for phase changes is always bigger than the one for the point wise noise in all cases.

Hence, It is critical to operate at a right coverage rate. For our hypothetical examples, it seems to be somewhere between 99% and 97%.

To sum up, modified Hausdorff distance has the following desirable properties:

- Through the fine-tuning of parameter τ, it has the flexibility to correspond points on a given signal with the neighboring ones on another given signal. Hence, it can tolerate shifts and delays, which often happen in time series data.
- It assigns distances with respect to only the amplitude differences between two given signals after correspondences as opposed to summing up all the differences at every time instant. So it is sensitive to two parameters: Amplitude and Phase shifts.
- It allows small deviations such as small point wise magnitude changes. It is robust to such noises.
- It is robust to unusual activities in a given time series: outliers.
- In addition, since it is mostly sensitive to Amplitude changes and Phase shifts and less sensitive / robust to other effects, it tries to capture the space of signals in 2 dimensions only. As a result of this, for the space of signals which can be parameterized through amplitude and phase shifts, it can capture the underlying topology of the space of such signals to a greater degree when compared to standard Euclidean based distance measures. And interestingly, for the space of such signals, i.e, signals that can be parameterized through Amplitude and phase shifts, using Hausdorff distance also means an effective dimensionality reduction since it is designed here on purpose for such signals.

Commonly used metrics on fMRI data and the Hausdorff metric:

Almost all clustering methods involve three steps: (1) Preprocessing of the raw data and feature extraction, (2) using the right distance or similarity metric, (3) and using the right clustering algorithm. In this thesis, we combine the first two steps because one can always embed the first step in the distance calculation either explicitly or implicitly. It is well known that a clustering algorithm can be as good as the distance or similarity metric used. For aforementioned properties, we think Hausdorff metric is a good choice for fMRI data clustering. To understand this, in this section we compare Hausdorff metric with standard metrics such as Euclidean distance, Mahalonobis distance, Radial Basis Function (rbf) similarity, cosine similarity and also the ones which are frequently used in fMRI analysis.

FMRI clustering has been extensively used for the identification of active regions in the brain as well as constructing a neural connectivity map. Most of them work either directly or through characteristic features such as the cross correlation with the experimental impulse train signal or again cross correlations between signals in the phase of defining the distances or similarities [19], [56], [70], [71], [74], [75]. For a couple of examples, in [19], the cross correlation function of a given time series (in our case this time series is the estimated hemodynamical time series, but cross correlation in these papers are directly used on raw fMRI signals, so we denote it as F instead of D) with the underlying impulse train (the stimulus) is used as features and then hierarchical clustering is applied. Basically the transformation in (3.54) is:

$$Z(n) = \sum_{m} F(m)s(n+m) \tag{3.54}$$

where F is the fMRI signal of interest and s is the underlying impulse train of the experiment which takes zero value when the subject is at rest and 1 when he is given a

stimulus, and *Z(n)* is the cross correlation function. In [19], *Z(n)* is used as the extracted features as well as the delay at which the maximum of *Z(n)* is observed. Based on these features, a clustering algorithm is applied. In this approach, the delay between the impulse train and the fMRI signal is carefully extracted and used. However, this approach requires the knowledge of the impulse train which is something we do not have in hand, and thus it is a signal we want to avoid in this thesis leading to a completely unsupervised method.

In [19], cross correlation is used as features, and the clustering based on the features uses the standard Euclidean distance. In another work [70], inspired by [19], the following distance measure is defined based on the cross correlation features: Given two fMRI signals, F_1, F_2; $D(F_1, F_2) = P_{12} + L_{12}$ where P_{12} is the peak difference in the corresponding cross correlation functions Z_1 and Z_2. And similarly, L_{12} is the delay difference in the corresponding cross correlation functions again. Here, as P_{12} decreases then the distance also decreases which makes sense because then also F_1 and F_2 gets closer to each other. And at the same time the delay between the two is also considered so that as the delay increases the distance also increases through L_{12}.

As a result, this approach also requires the knowledge of the impulse train. Another issue is that, considering three time series: $F_1(n), \quad F_2(n) = -F_1(n), \quad F_3(n) = 0.$ $F_1(n)$ and $F_2(n)$ is more similar to $F_3(n)$ than with each other, because $P_{12} = |P_1 - P_2| = 2|P_1|$ while $P_{13} = P_{23} = |P_1|$. This is counter intuitive, since a positive signal should be considered more similar to a negative signal. Furthermore, the metric defined above implicitly assumes that the temporal extent of the hemodynamic response and the underlying impulse train are equal which is not necessarily true. However, this kind of a metric might be used for a data reduction tool to which we will return later. So we can call this kind of metrics as 'semi model based metrics'.

On the other hand, in [56], a signal to signal distance measure is defined again based on cross correlations as a modification to the metric above:

Given two fMRI signals: F_1 and F_2

$Z_{12}(n) = \sum_m F_1(m)F_2(n+m)$, $P_{12} = \max(|Z_{12}(n)|)$, $L_{12} = \arg\max(Z_{12}(n))$.

And the distance, $D(F_1, F_2) = -P_{12} + L_{12}$

When compared to the semi model based metrics, this does not require the knowledge of the underlying impulse train. However, it still incorporates with the cross correlation. This time, as P_{12} increases, the distance value decreases since a greater P_{12} means a greater similarity or a stronger correlation between $F_1$and F_2. And similarly, the delay between the two again is considered in the distance function through L_{12}.

So these kind of metrics or slightly different versions of them are commonly used in fMRI data analysis. There are two common properties here to be noted: (1) the delay between given two fMRI signals are carefully extracted and it is incorporated in the distance function and secondly, (2) after the delay is compensated, the similarity between the two signals is measured by the correlation, or by the inner product. And finally a distance metric is constructed.

Comparing the modified Hausdorff metric with cross correlation based metrics, especially the first property above makes good evidence that the delay or misalignment is an important issue not only in the analysis of time series data in general but also in fMRI analysis. And hence it makes the modified Hausdorff metric a good choice since it has its own way to deal with the delay. Intuitively the idea behind the Hausdorff metric can also be posed as in (3.55):

Hausdorff distance:

$$d(F_1, F_2) = \sqrt{(\text{maxAmpDev}(F_1, F_2))^2 + (delay(F_1, F_2))^2} \tag{3.55}$$

Similarly, the idea of the cross correlation based metric is:

$$d(F_1, F_2) = -Peakof\ CrosCorrelation(F_1, F_2) + delay(F_1, F_2) \tag{3.56}$$

The fundamental difference between the modified Hausdorff metric and the cross correlation based metrics is that modified Hausdorff metric is only sensitive to the maximum amplitude deviation between the given two fMRI signals. Let us say the difference between the peak-to-peak amplitude difference is 1 (in magnitude units) given two signals, then the modified Hausdorff distance is only sensitive to this quantity which means that it is completely insensitive to any magnitude differences less than 1. Through this property, the modified Hausdorff metric can tolerate any kind of variations and noise at any time points except the ones where peaks are located. On the other hand, cross correlation based distances are sensitive not only to peak-to-peak properties. In order for two signals to become similar in the case of cross correlation based distances, their overall shapes should be globally correlative, meaning that, after the delay is compensated, the inner product between them should be large. For an experimental demonstration on real fMRI signals, we choose an fMRI signal, which has an ID number 608, which is shown in Figure 3.17.

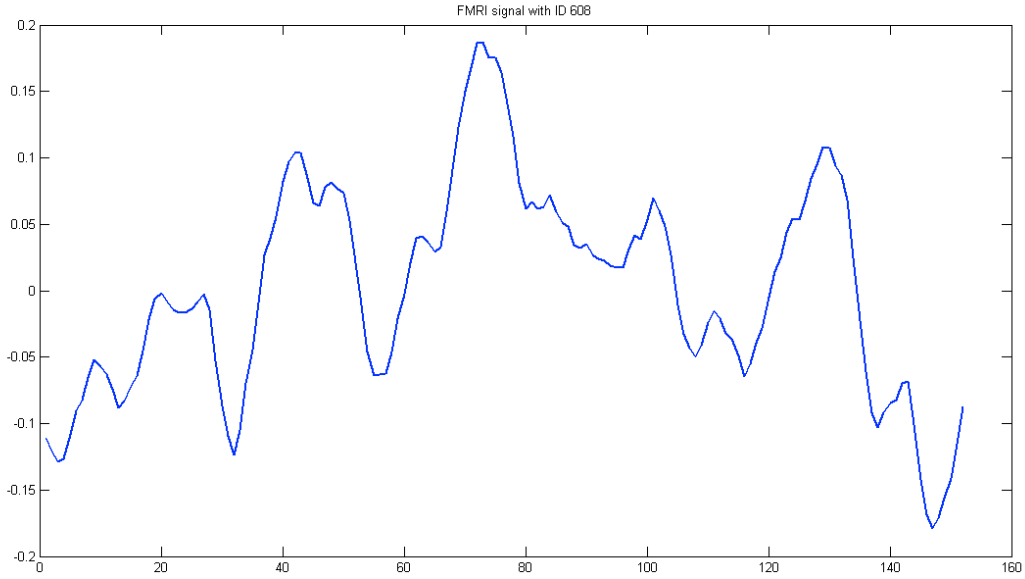


Figure 3.17 fMRI signal with ID 608

Then we find the nearest 10 neighbors of it (including itself) with respect to modified Hausdorff metric (with parameters $\alpha = 95\%$, $\tau = 0.01$) as well as the cross correlation based metric, which are:

Nearest 10 neighbors of the fMRI with ID: 608 w.r.t. modified Hausdorff metric = {608 607 609 611 660 664 551 663 661 553}

Nearest 10 neighbors of the fMRI with ID: 608 w.r.t. cross correlation based metric = { 608 607 609 605 611 660 604 553 664 551}.

For a discussion of these, some examples are shown in Figure 3.18:

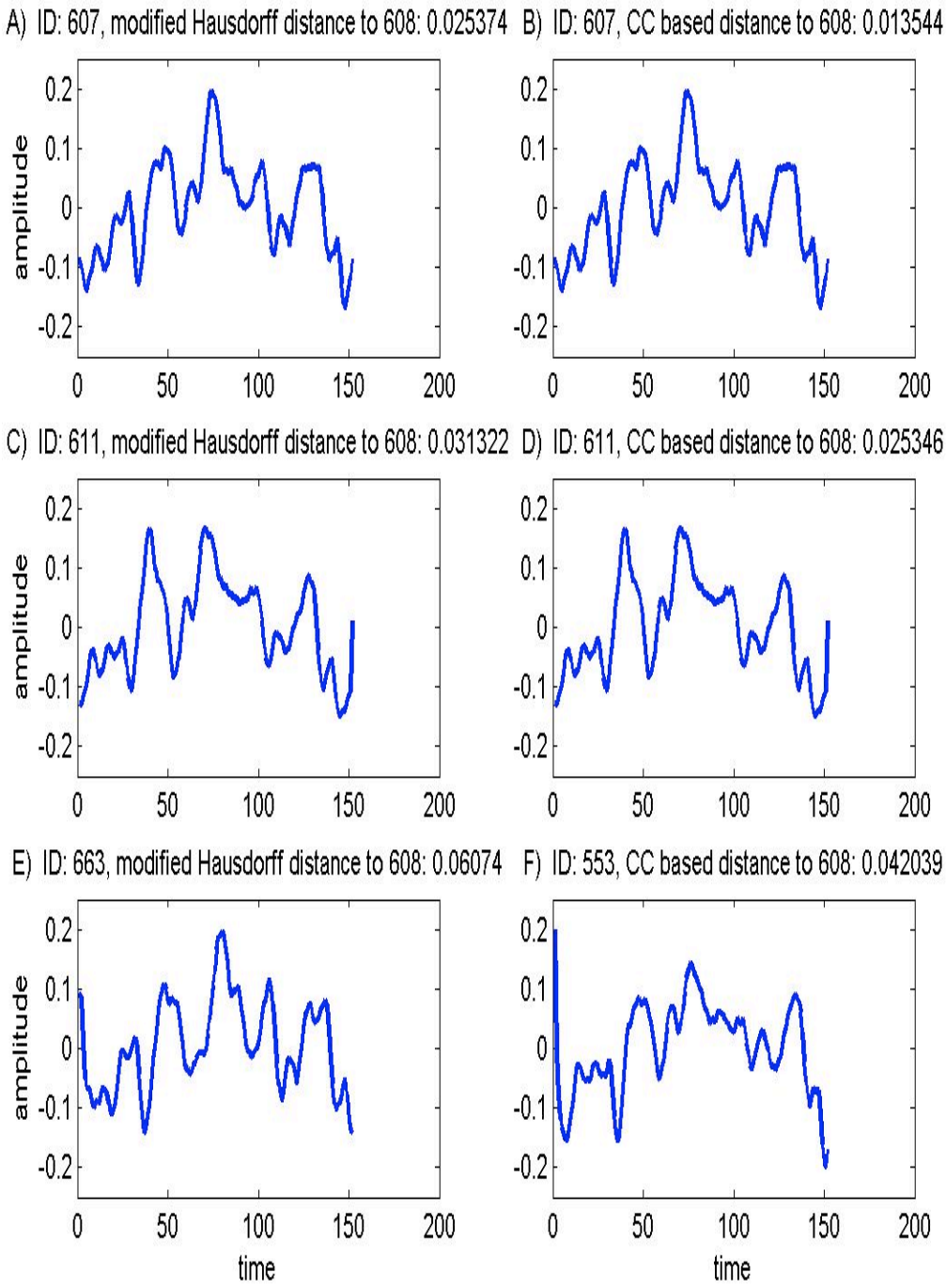


Figure 3.18 Cross correlation and modified Hausdorff distance between the fMRI with ID: 608 and its neighbors

Both metrics seem to be good at finding the similar signals among nearly 2000 fMRI signals given the query signal 608. In the Figure 3.18 it is shown a couple of examples from the neighborhood of the query signal with respect to both metrics. First to note here is that the modified Hausdorff metric, as expected, is not affected by the large deviations in magnitude values of a short duration. For instance, the initial value of signal 611 (Figure 3.18 C) is around -0.1 whereas the same value for 663 (Figure 3.18 E) is around 0.1. Note that the query signal (Figure 3.17) also starts with a magntiude around -0.1 and so eventhough the maximum magntiude-wise deviation between the signals 608(query) and 663 is at least 0.2 initially but still this does not last too long, they can still be detected as similar with the help of the ability to adjust the coverage rate through α parameter. And next, even though the signal 663 is almost 10 time points ahead of the query signal, the modified Hausdorff metric is still able to catch the similarity. We can make the similar arguments for the cross correlation based metric; hence, with respect to the query signal, both metrics seem to perform similarly.

However, the signals 661 (Figure 3.19E), 663 (Figure 3.19C) are not included in the neighborhood with respect to the cross correlation based metric, and the signals 604 (Figure 3.19F) and 605 (Figure 3.19D) are not included in the one with respect to the modified Hausdorff metric. These signals are as in Figure 3.19:

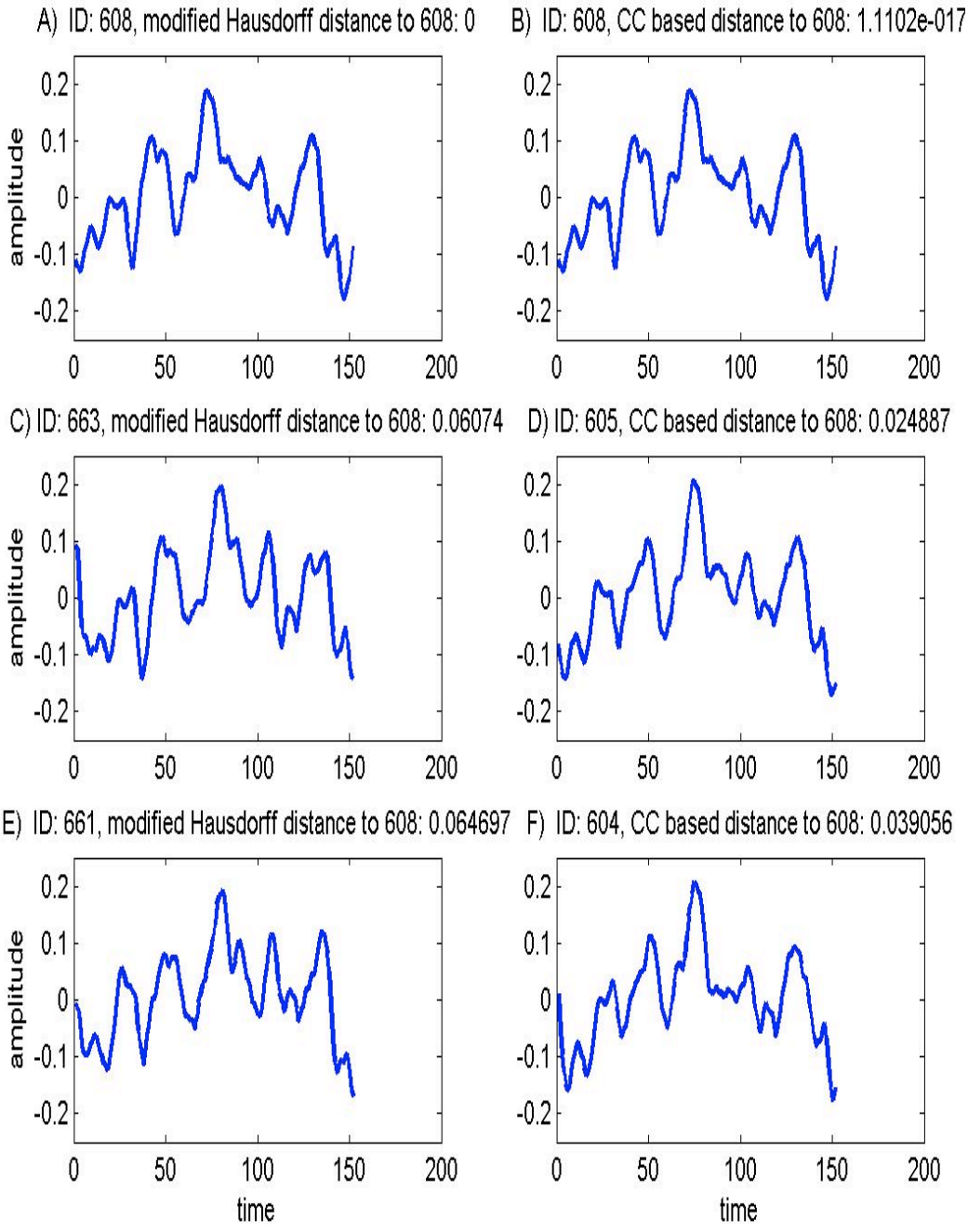


Figure 3.19 Different neighbors of fMRI with ID 608 wrt modified Hausdorff and cross correlation metrics

Considering the signals 663 (Figure 3.19 C) and 661 (Figure 3.19 E), they are not detected by the cross correlation (CC) based metric mostly because the shape of the signals in the time intervals around [55 65] and [80 100] do not match the corresponding parts of the query signal (Figure 3.17). CC based metric is sensitive to such variations. On the other hand, the modified Hausdorff metric can tolerate such variations as long as the maximum deviation between the two signals (maximum of length of all correspondences) is not affected. As for the signals 604 (Figure 3.19 D) and 605 (Figure 3.19 F), they are not detected, this time, by the modified Hausdorff metric because in the time interval [35 50] there is too much magnitude deviation between them and the query signal 608. Hence, this takes them out of the 10-neighborhood calculated by the modified Hausdorff metric.

As a result, the two metrics are behaving very similarly but if taken a closer look, the fact that Hausdorff distance is more sensitive to amplitude changes and the cross correlation is more sensitive to the overall shapes of the signals through correlations, becomes appearant.

Hausdorff Metric in Comparison to Standard Metrics

In this section we compare the modified Hausdorff metric to some of the well-known metrics:

- **Euclidean Distance(L_2-distance):**

  - Given two signals F_1 and F_2, the Euclidean distance is defined as;

$$dE(F_1, F_2) = \|F_1 - F_2\|_2 \tag{3.57}$$

  - Since it assumes 1-1 correspondences between ordered vector components, it is overly sensitive to shifts and delays in the signals.
  - It is also sensitive to outliers such as some unusual activity of a short duration with respect to the general shape of a given signal.
  - We can also extend the properties above in general for L_p-distances,

$$dL_p(F_1, F_2) = \sqrt[p]{\sum (F_{1i} - F_{2i})^p} \tag{3.58}$$

since, for all p, it still does not have the ability to tolerate delays and outliers since the generalization through p does not bring any capability to cover such delays or outliers.

- **Generalized Euclidean Distance(Mahalanobis distance):**

Through linear transformations of the data, a fairly large class of metrics can be obtained by defining the following generalized distance:

- Given two signals F_1 and F_2, the Mahalanobis distance is defined as;

$$dM(F_1, F_2) = \sqrt{(F_1 - F_2)^T \Sigma (F_1 - F_2)} \tag{3.59}$$

where Σ is a symmetric positive definite matrix. If it is used as the identity matrix, generalized Euclidean distance precisely turns out to be the standard Euclidean distance. If it is a diagonal matrix, then this metric first scales the signals in each dimension (or at each time instant) with the corresponding diagonal element, and then calculates again the standard Euclidean distance. And in general, since Σ is positive definite, then there exists T such that $\mathbf{\Sigma}=\mathbf{T}^T\mathbf{T}$. Then,

$$dM(F_1, F_2) = \sqrt{(F_1 - F_2)^T \Sigma (F_1 - F_3)} = \sqrt{(TF_1 - TF_2)^T (TF_1 - TF_3)} = dE(F_1', F_2') \tag{3.60}$$

where $F' = TF$ which defines a linear transformation through the matrix T.

Hence, in general, Generalized Euclidean Distance is just the standard Euclidean after some linear transformation.

- The data covariance is often used as Σ. In this case the corresponding linear transformation uncorrelates the data and makes it of unit variance in each dimension. In terms of distances, this might be a better choice because it adaptively finds the principal directions in the data and then it calculates the Euclidean distance according to the directions and corresponding variances. For instance, from a point to another one on a

direction on which there is large variance, this distance then is relatively small compared to the ones having small variance.

- Despite good properties, however, it is still not suitable for fMRI clustering. Firstly because, it requires the covariance estimation. In general fMRI data have high dimensionality, say, 150. Then ((150x150+150)/2)=11325 parameters need to be estimated. And most of the time a large enough sample size is not available in order to have a good estimation. Secondly, this metric also does not address the delay and outlier issues because it brings a generalization through linear transformations only. On the other hand, we want to discard some unknown parts of the signals intelligently if there exists an outlier and compensate for possible delays. These two together already require a nonlinear process and would increase the complexity of the method.

- **Gaussian Similarity:**

As opposed to measuring 'dissimilarity' between two data points by defining a distance function, one can also use a 'similarity' measure as they are very close notions being the inverse of each other. One of the very well-know similarity measures is Gaussian Similarity is defined as (3.57):

$$simG(F_1, F_2) = \exp(-g.dE^2(F_1, F_2)) \tag{3.61}$$

where g is the bandwith parameter controlling how fast the similarity decays from one point to another as they get further from each other.

- The quality of this similarity measure is dependent not only on the bandwidth parameter but also on the distance funtion whose exponential is taken. In this example, and in its mostly used form, standard Euclidan metric is used. Hence, unfortunately, this kind of measures do not help in terms of delays and outliers.

- **Cosine Similarity:**

This is another commonly used similarity mesure in data analysis. Given two data points, it is basically cosine of the angle between them.

$$simCos(F_1, F_2) = \frac{F_1^T F_2}{\|F_1\| \|F_2\|} \tag{3.62}$$

- Cosine similarity turns out to be just the inner product between the data points when they have unit norm. And in this case, interestingly, it is precisely the cross correlation at zero'th shift. However, since we can actually observe shifts in the data as we already mentioned, this kind of a similarity measure is also not a good choice for our approach. If the delay is compensated and then the cosine similairty is calculated, then this might do a decent job. Though, then it would be nothing but a different version of the cross correlation based metric which has been already analyzed versus the modified Hausdorff metric.

For an experimental comparison of these metrics, we choose three fMRI signals from nearly 2000 fMRI's. For the calculation of mahalanobis distance, we use the empirical data covariance. A summary of our findings are in Figure 3.20:

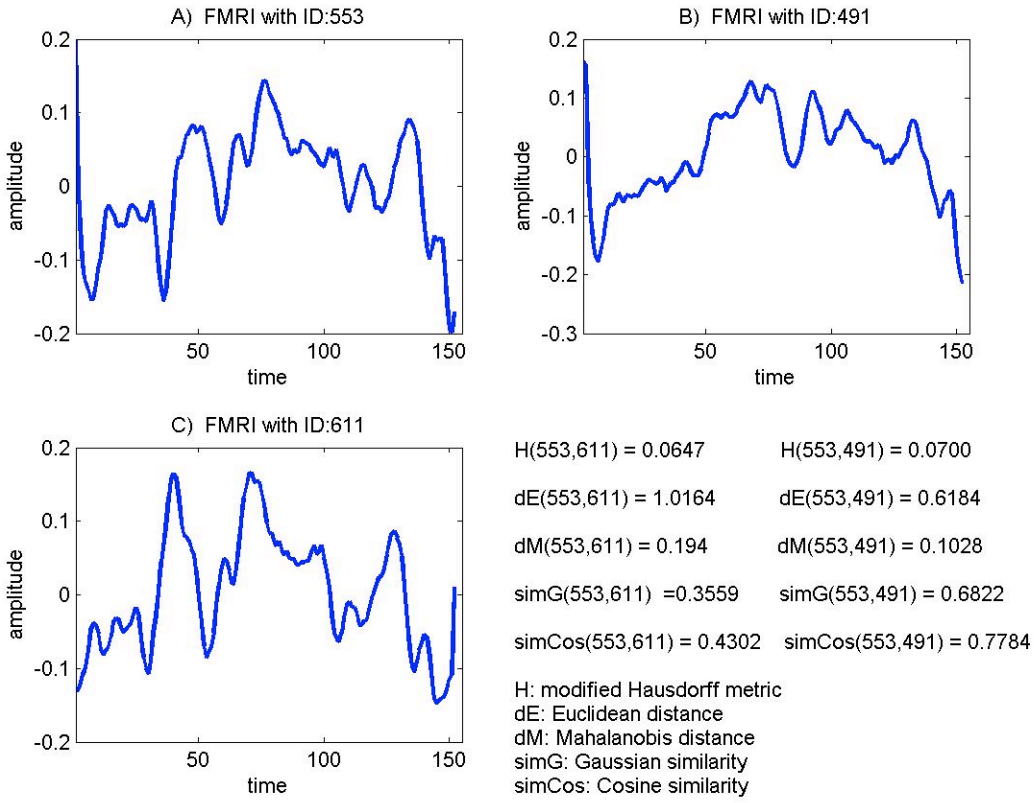


Figure 3.20 Comparison of the metrics

Visually speaking, FMRI_553 (Figure 3.20 A) and FMRI_611 (Figure 3.20 C) are similar to each other but FMRI_553 is 10 time points ahead of FMRI_611, moreover, there is also an outlier effect in FMRI_553 which is that it starts with a very high magnitude, 0.1. On the other hand, FMRI_491 (Figure 3.20 B) is shape-wise significantly different from the other two. Except the modified Hausdorff distance, FMRI_553 turns out to be more similar to FMRI_491 than it is to FMRI_611 for all other metrics and similarity measures which is incorrect due to the delay between them and the outlier in the beginning of FMRI_553. On the contrary, modified Hausdorff metric succesfully satisfies our visual expectations and detects the shape-wise similarity between FMRI_553 and FMRI_611 with the help of its capability of coping with possible delays and its robustness to outliers as $H(553,611) < H(553,491)$.

So, according to our discussions and comparisons, the modified Hausdorff metric is readily a powerful choice for fMRI data analysis and it has a close relationship with the Cross Correlation Based metrics in terms of the way they deal with possible lags in the observed fMRI signals. Hausdorff metric matches the time instants with the ones in another given signal by finding the amount of shift. On the other hand, Cross Correlation Based Metrics matches the signals by finding the possible shifts in between two given signals. From a perspective of computational complexity, since Hausdorff metric calculates all possible distances between every pair of fMRI magnitudes at time instants, it has a complexity of $O(N^2)$ where N is the dimension of fMRI data. However, if we expect at most z amount of delay, then the complexity for a cross correlation operation would be O(zN). So instead of matching the time points, via cross correlation first we estimate the possible delays, and then we apply modified Hausdorff metric. Also throughout our work we do not use stimulus pattern (since it is unknown to us in our problem), which is needed for the cross correlation metric. From these points of view modified Hausdorff distance is chosen as distance measure.

Given two fMRI signals: F_1, and F_2, if the estimated delay between them is $delay(F_1, F_2)$ and τ is our scaling parameter, then the distance measure that is used in our clustering algorithm is:

$$d_{fMRI}(F_1, F_2) = \sqrt{H_\alpha^2(F_1, F_2) + \tau^2 delay^2(F_1, F_2)} \tag{3.63}$$

3.4 Spectral Clustering after MAP Blind Deconvolution

After calculating the modified Hausdorff distances between the pairs within the fMRI data set, we use them for constructing the similarity graph on data and so for generating the matrix W in our algorithm, which holds the edges and edge weights for each pair of graph nodes. There are several ways proposed for this part, such as generating fully connected graphs and partially connected graphs. For fully connected graphs the most common method is that each node is connected to every other node and the weight for each node is assigned according to a Gaussian kernel using Euclidean distance, i.e, $W(i,j) = \exp(-g * \left\| d_i - d_j \right\|^2)$. This is also called similarity matrix. As for the partially connected graphs, the most common way is taking a neighborhood approach and masking the similarity matrix, i.e,

$$W(i,j) = \begin{cases} \exp(-g * \left\| d_i - d_j \right\|^2), & \text{iff } \mathrm{d}_i \text{ is in the } k\text{'th neighborhood of } \mathrm{d}_j \\ 0 \quad , & \text{otherwise} \end{cases}$$

$$W(i,j) = \max(W(i,j), W(j,i))$$
$$W(j,i) = W(i,j)$$

Taking the maximum of two edge weights is to just make the similarity matrix symmetric. Using the minimum is also possible. Instead of having a real valued similarity matrix, using a binary valued similarity matrix is also reasonable choice, i.e,

$$W(i,j) = \begin{cases} 1\ , & \text{iff } \mathrm{d}_i \text{ is in the } k\text{'th neighborhood of } \mathrm{d}_j \\ 0\ , & \text{otherwise} \end{cases}$$

$$W(i,j) = \max(W(i,j), W(j,i))$$
$$W(j,i) = W(i,j)$$

In this work, we used a binary valued similarity matrix with respect to modified Hausdorff distance. That is to say, we detected whether a point d_i is in the k'th neighborhood of d_i by using the modified Hausdorff distance. Once having the similarity matrix constructed, then spectral clustering is solving the generalized eigenvalue problem $Lv = \lambda Dv$ where L is the Laplacian and D is the diagonal matrix computed via similarity matrix W. Then, the eigenvectors returned by the generalized eigenvalue problem are sorted in ascending order with respect to the corresponding eigenvalues and, depending on the desired number of cluster that one want to find in a given data set, first nclass of them give the spectral mapping:

Given the eigenvectors as:

$$V = \begin{bmatrix} | & | & | \\ v_1 & v_i & v_m \\ | & | & | \end{bmatrix}_{mxm}$$

where $\lambda_1 < \ldots < \lambda_i < \ldots < \lambda_m$,

so the spectral mapping of a data point given nclass is

$$\mathrm{d}_i \in \Re^{\dim x 1} \rightarrow \mathrm{d}' = \begin{bmatrix} V_{i1} \\ V_{i,j} \\ V_{i,nclass} \end{bmatrix} \in \Re^{\mathrm{nclass\,x\,1}}$$

In the end clustering is finalized by applying EM clustering in the spectrally transformed space. Here the reason using EM instead of using k-means is that EM is a more sophisticated algorithm and is capable of detecting nonlinear boundaries between clusters.

Algorithm

In what follows, we summarize our complete algorithm. The preprocessing step was already explained in Section 3.2. The details of the algorithm are going to be discussed along with the experimental results in Chapter 4.

Algorithm (using MATLAB notation):

- Input:
  - fMRI data: $\{r_i\}_{i=1}^{N}, r_i \in \Re^{1xT}$
  - Parameters:
    - Blind Deconvolution parameters: κ, p
    - Hausdorff distance parameters: τ, α
    - Spectral Clustering parameters: $k, nclass$
- Processing of fMRI:
  - Extract each hemodynamic: $r_i \leftarrow \text{get\_hemodynamic}(r_i)$
  - Normalization:
    - $r_i \leftarrow r_i - mean(r_i)$
    - $r_i \leftarrow r_i / norm\ (r_i, 2)$
  - Drift Elimination
    - $r_i \leftarrow \text{Eliminate}(\text{drift})$
- Compute the distance matrix:
  - $s_i \leftarrow [r_i; \tau * (1:T)]$
  - $\mathbf{d}(i, j) \leftarrow \text{Haus\_dist}(s_i, s_j; \alpha, \tau)$
- Spectral Estimation:
  - Construct the graph matrix $\mathbf{W}$
    - $\mathbf{W}(i, j) \leftarrow 1 \Leftrightarrow s_i$ is in the k'th neighborhood of s_j
    - Symmetrization:

- $\mathbf{W}(i,j) \leftarrow \max(\mathbf{W}(i,j), \mathbf{W}(j,i))$
- $\mathbf{W}(j,i) \leftarrow \mathbf{W}(i,j)$

  - Compute the matrices: $\mathbf{L}, \mathbf{D}$
    - $\mathbf{D} \leftarrow diag(sum(\mathbf{W},2))$
    - $\mathbf{L} \leftarrow \mathbf{D} - \mathbf{W}$
  - Compute the eigen map
    - $\mathbf{EV} \leftarrow eig(\mathbf{L},\mathbf{D})$ (eigenvectors are sorted w.r.t eigen values in ascending order)
    - $x_i \leftarrow \mathbf{EV}(i,1:nclass)$

- Clustering
  - EM: $id\_EM \leftarrow EM(\mathbf{X},nclass)$

CHAPTER 4

4 EXPERIMENTS AND RESULTS

There are two types of experimental designs commonly used in fMRI: blocked design and event-related design. In our work both the simulated and the real fMRI data adheres to the block design experiment paradigms. Before going further in the results of our experiments, we will mention what is the meaning and difference of these two paradigms in order to make more realistic comments and understand the meaning of the outputs in our algorithms. After experimental paradigm explanations we will give and discuss the results of our algorithm.

Blocked designs were the earliest type of experimental design used for fMRI research and remain important today. In a blocked design, there are two conditions, the experimental condition and the control condition. The experimental condition is the task that the researcher tries to test which is known as task block. These tasks may involve finger tapping, lip pursing, and toe curling, illustrations of lights, patterns, photographs, word rhyming, word generation, decision making in response to a stimulus that is delivered visually or auditory. In the control condition, the stimulus is either not present

at all or it is much less evident which is known as rest. The task and rest blocks are alternated, allowing the hemodynamic response to rise and saturate. Statistical comparison of the two states is done, and any background neuronal activity occurring during the rest block is subtracted from the activity elicited by the independent variable. The task blocks may be the same each time or may consist of different tasks, and the time between the blocks can be varied as needed. When blocks are too short, there is very little difference between the task and rest blocks. Longer blocks give the best separation between peak hemodynamic response and resting state. However, when blocks are too long, low frequency noise becomes a big problem. Blocked designs are very good at detecting active voxels, but not very good at estimating the timing of the hemodynamic response. Blocked designs are simple to set up and understand, and they provide strong, statistically relevant results. Blocked designs are the most frequently used design for clinical fMRI. An example of an fMRI time course using a blocked design is provided in Figure 4.1.

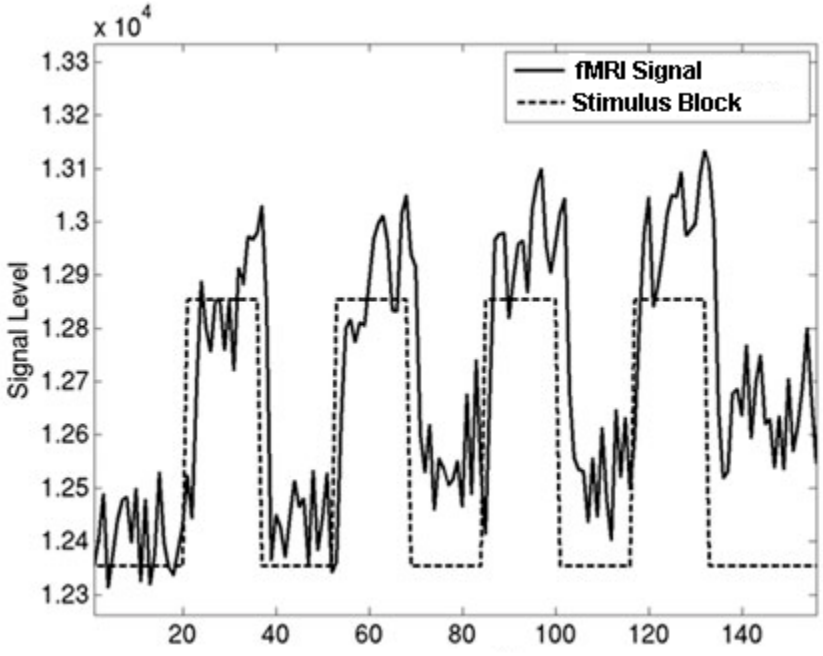


Figure 4.1 fMRI signal generated using Block-Design Experiment

There is also a special and advanced type of block design, which is called **categorical block design.** By means of categorical block design, activation in one task is compared to that in another task. Categorical designs assume that cognitive processes can be dissected into sub-cognitive processes and that one can add or remove cognitive processes by “Pure insertion”. Pure insertion assumes that one can add or remove cognitive processes without influencing others. Activation in one task is compared to that in another task considering the fact that the neural structures supporting cognitive and behavioral processes combine in a simple additive manner using categorical block design paradigm. It gives the opportunity of testing multiple categorized hypotheses. Several hypotheses are tested, asking whether all the activations in a series of task pairs are jointly significant.

Event-related design, is the other major type of experimental design used in fMRI, well suited for stimuli that generate brief bursts of neural activity. An example of fMRI using an event-related design is a flash of bright light that elicits a burst of activity in the occipital cortex. These stimuli are known as "events". Event-related designs are very good at estimating the timing of the hemodynamic response and can be optimized to infer similar activation information as in blocked designs. An example of an event-related design is shown in Figure 4.2.

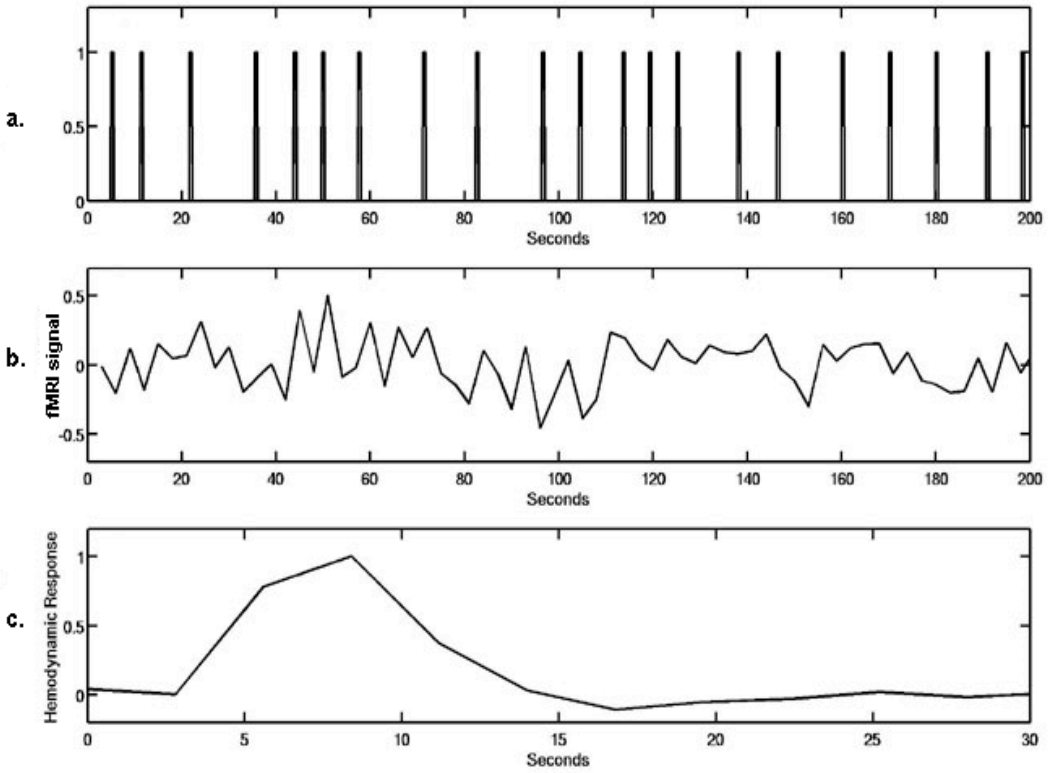


Figure 4.2 Example of Event-Related Experiment

a. Impulses of stimuli b. fMRI signal c. Hemodynamic response function

In this chapter initially we will give the results of hemodynamic response estimation. These estimated HRFs give the characteristics of the related voxel. Then changing the convolution filter parameters we obtain smooth time series that contain the hemodynamical time series. These time series include the hemodynamical response to each stimulus block and gives the instant information how the subject responded to the administered impulse pattern. These hemodynamical time series are then used as input of our clustering algorithm. Finally the results for clustering will be discussed.

During the experiments we used one set of simulated data and two sets of real data. For the simulated data set we used Balloon Model [80], [81], [82] with parameters ε=0.5, τS= 0.8, τf= 0.4, $\tau 0$= 1, α= 0.2, E0 = 0.8, V0 = 0.02.

First real fMRI data set ”Data27” is obtained from a block design experiment. In this block design experiment, a classical finger tapping paradigm is used, with 60 time points collected in 3 cycles which contained 10 samples for each ON or OFF periods through the echo planar imaging protocol.

Second real fMRI data set is obtained from a categorical block design experiment. In this experiment, fMRI data is obtained from a 1.5T Siemens scanner. It is an fMR adaptation paradigm investigating subtle effects in face processing. This fMRI data consists of 177 time points with 6 cycles.

Active and passive voxels in both real datasets are classified beforehand via general linear model, which served as ‘ground truth’.

4.1 Hemodynamic Response Function Extraction

Recalling our convolution regarding the observed fMRI signals, we have the model in (4.1):

$$r(t) = d(t) \otimes k(t) + n(t) \tag{4.1}$$

where,

r(t): Observed signal, fMRI.
d(t): Hemodynamic response function.
k(t): Convolution filter (stimulus).
n(t): Additive White Gaussian Noise (AWGN)
t: Discrete time

In the fMRI literature, Hemodynamic Response Function (HRF) is known as the impulse function of brain voxels. Hence, ideally, fMRI signals are modeled as response of brain voxels to a given stimulus, such as an impulse train like a square wave which can be posed as the convolution of HRF with the stimulus as in (4.1) with noise added on. In the methodology chapter, we explain our blind deconvolution algorithm and study the case for estimating the hemodynamical time series where *k(t)* is used as a convolution filter of a small support. If we use *k(t)* of full length or close to full length, we hope to recover the stimulus as *k(t)* and extract HRF as *d(t)*. Ideally, HRF is expected to have a characteristic shape as shown in Figure 4.3:

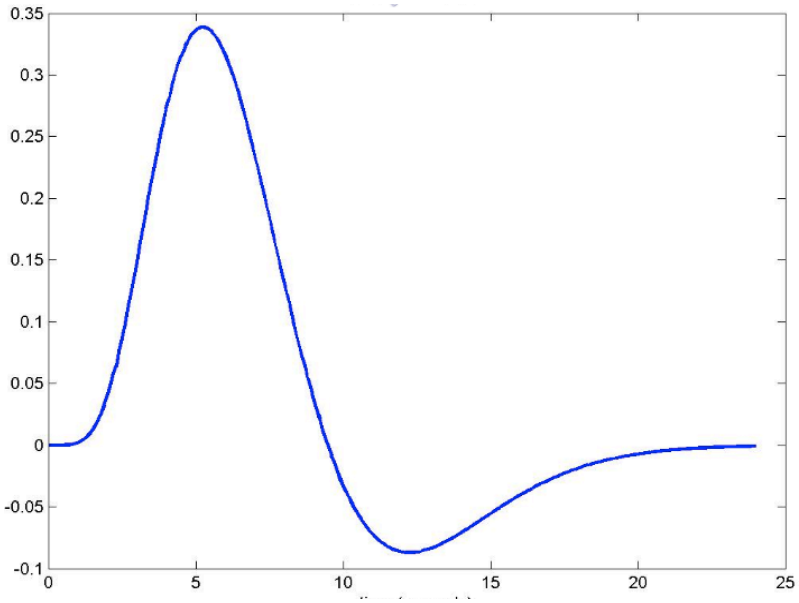


Figure 4.3 Hypothetical Hemodynamic Response Function

A typical HRF as shown in Figure 4.3 exhibits a rise to a peak around 5-6 sec followed by an undershoot, which lasts until 10-12 sec and then a recovers around 15-18 sec. The duration for HRF to reach its peak is reflected as a delay in the BOLD response, or in fMRI time series due to the convolution. And assuming that there is a sufficient time between impulses in a given stimulus, the resulting fMRI time series should have peaks as many as the number of single impulses or task blocks.

To test our HRF extraction, first we used the simulation data, which is based on the Balloon Model [80], [81], [82]. The parameters for the simulation of BOLD signal change is the same as in [82], ε=0.5, τ_S= 0.8, τ_f= 0.4, τ_0= 1, α= 0.2, E_0 = 0.8, V_0 = 0.02. Then we applied our algorithm on both real data sets.

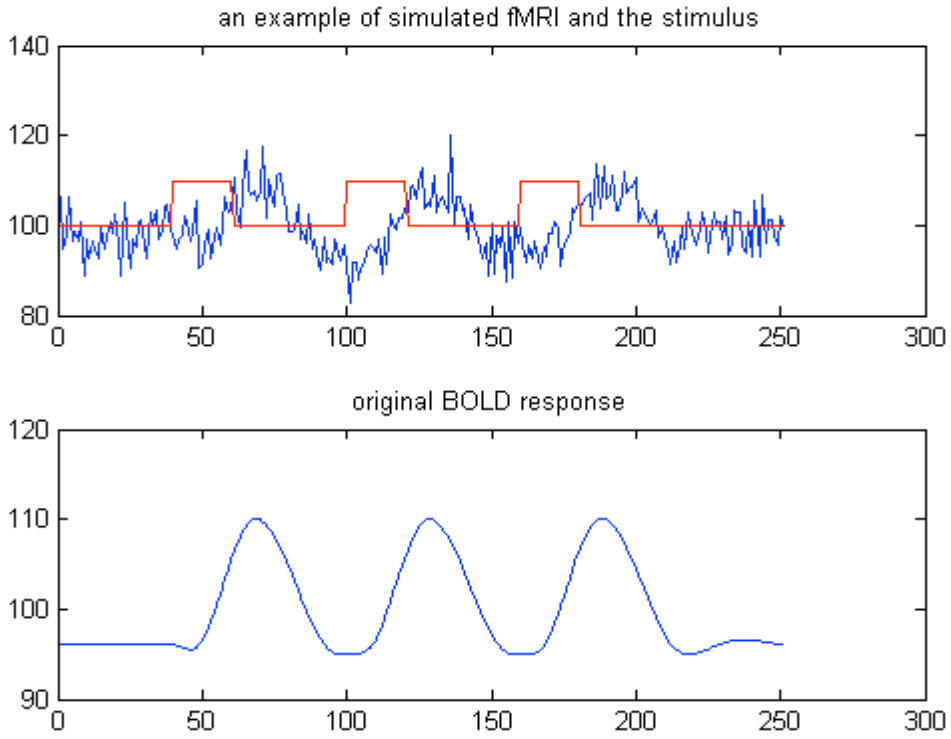


Figure 4.4 An example of simulated fMRI and its stimulus pattern and its original BOLD response

We apply stimuli blocks between the time intervals of [40, 60], [100, 120], [160, 180] for generating simulation data. In Figure 4.4 first thing to note is that the BOLD response of the fMRI signal is a little delayed with respect to the corresponding stimulus. This is actually due to the rising time of HRF. Secondly the base level of the BOLD response is around 96 and the value it takes at time instant 100 is around 95, the difference in between the peak and minimum value that BOLD response takes is precisely due to the undershoot of HRF.

We run our MAP blind deconvolution algorithm to estimate the Hemodynamic Response Function, for an example, on this example with the parameters:

- $\kappa = 10^{-5}$
- $p = 200$ (should be sufficiently sufficiently large)

Here, we choose a very small number for κ, which controls the smoothness of estimated HRF. In order to avoid any large swings on HRF, we should set it small. And also we choose a sufficiently large number of the convolution filter length since we aim to capture the whole stimulus pattern as our convolution filter. Note that the last time point at which the true stimulus has an impulse, is 180, so *p* should be at least 180. Under these settings, we successfully estimate the stimulus when only the observed fMRI is given. Our findings are illustrated in Figure 4.5:

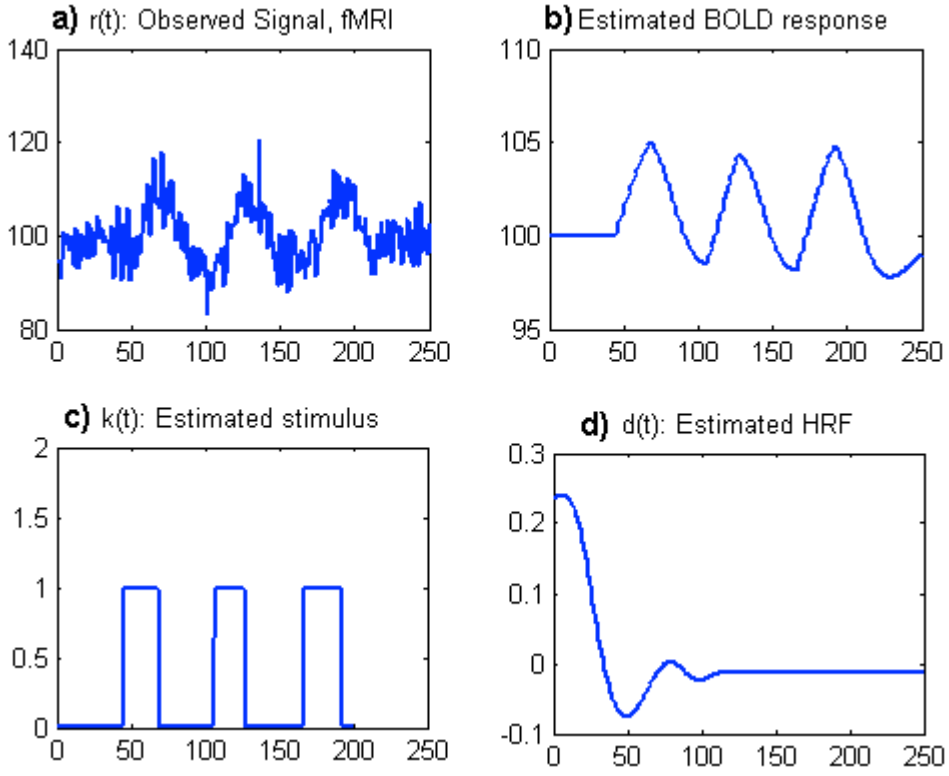


Figure 4.5 Estimated HRF, estimated stimulus and their convolution "Estimated BOLD Response" given a simulated fMRI signal with $\kappa = 10^{-5}$ and p=200

As shown in Figure 4.5, we extract the unknown stimulus very nicely (Figure 4.5c), which seems almost perfect except that it is shifted rightward by an amount of 5 time points. As for our estimated HRF (Figure 4.5d), we can argue that our algorithm

performs well except that HRF estimation misses the rise to the peak of the ideal HRF. Otherwise, it has a nice undershoot and it also settles around zero. To understand why it misses the rise to the peak, we should concentrate on our smoothness constraint that we impose on HRF. Basically, a quick rise to the peak violates our smoothness constraint in the iterative optimization phase of our MAP blind deconvolution. In order for our deconvolution to also find the rise to the peak in the HRF, it has to locate the stimulus at its exact locations, which should be shifted leftward with respect to the one shown in Figure 4.5 by amount 5 time units. Then, the approximation error of the observed signal in our model by the convolution would not change much. On the other hand, the HRF then would have a larger sum of first order derivatives, which is further penalized in our optimization. Hence, our algorithm basically chooses the stimulus at its original location and misses the rise to the peak in favor of lower derivatives. In this case one can immediately think to lower the smoothness parameter κ, one example of this idea is as follows with $\kappa = 10^{-4}$:

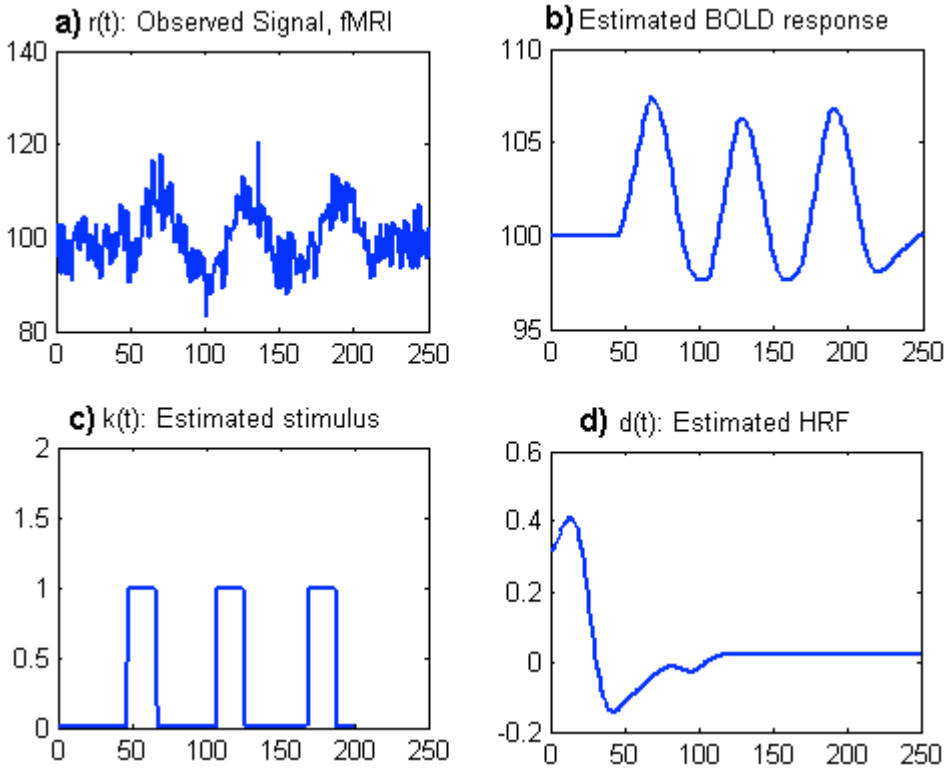


Figure 4.6 Estimated HRF, estimated stimulus and their convolution "Estimated BOLD Response" given a simulated fMRI signal with $\kappa = 10^{-4}$ and p=200

In this case, we have a little better catch of the rise to the peak in the estimated HRF (Figure 4.6d) which, though, still does not seem satisfactory enough. Note that this time we have a better estimate for the BOLD response (Figure 4.6b). Hence, we better increase the smoothness parameter $\kappa = 10^{-3}$:

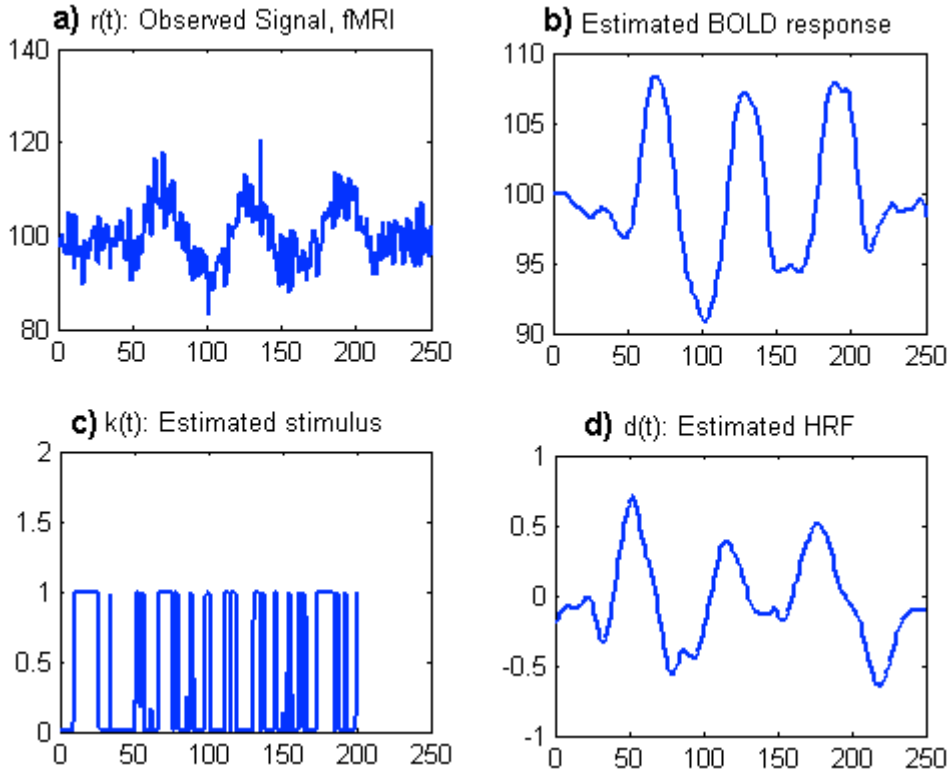


Figure 4.7 Estimated HRF, estimated stimulus and their convolution, "Estimated BOLD Response", given a simulated fMRI signal with $\kappa = 10^{-3}$ and p=200

However, Figure 4.7 shows that further increasing the smoothness parameter does not actually help. The algorithm after a critical value for κ between 10^{-4} and 10^{-3} starts estimating the HRF and stimulus in favor of a better data fitting which means that algorithm starts fitting to noise as well. For that reason, HRF and stimulus estimation are not satisfactory in this case.

Next we do a similar analysis on the categorized block design real data set. Figure 4.8 shows the real fMRI data with voxel ID_70 from this data set together with the stimulus:

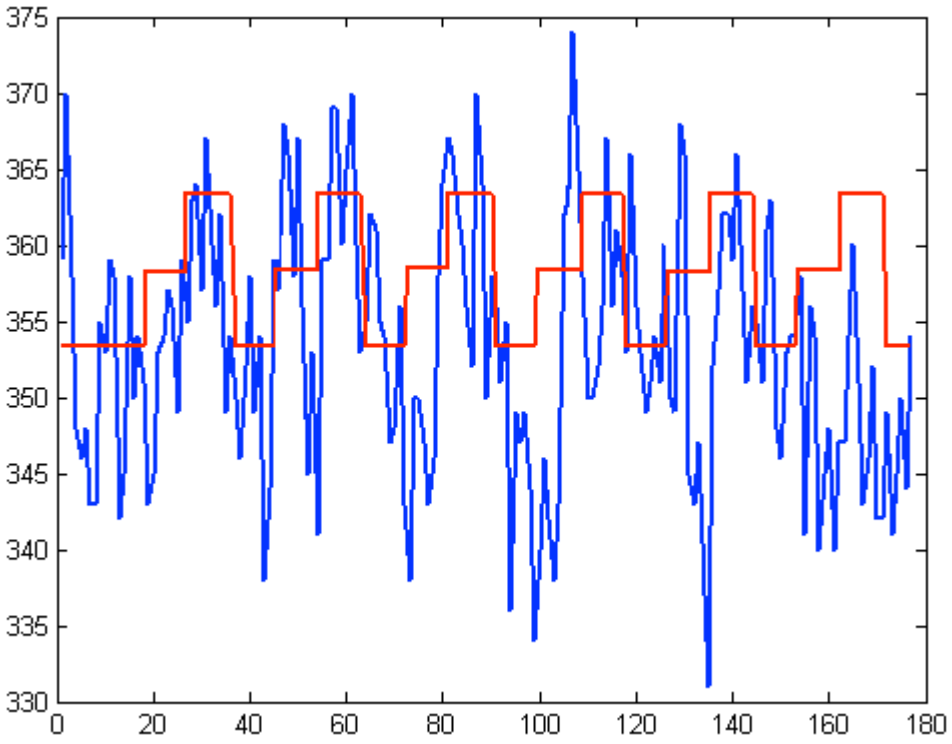


Figure 4.8 The real fMRI with ID_70 and its stimulus pattern

The stimulus of this experiment actually consists of impulse blocks each of which has two different successive categories. As a result of this type of design, in each block, fMRI signals tend to have two sub-blocks, which show itself in this example as well. Note that for the blocks 2, 3, 4, and 5, two separate peaks are visible.

When we run our MAP blind deconvolution on this fMRI signal, we estimate, as before, HRF, BOLD response and stimulus which are all shown in Figure 4.9:

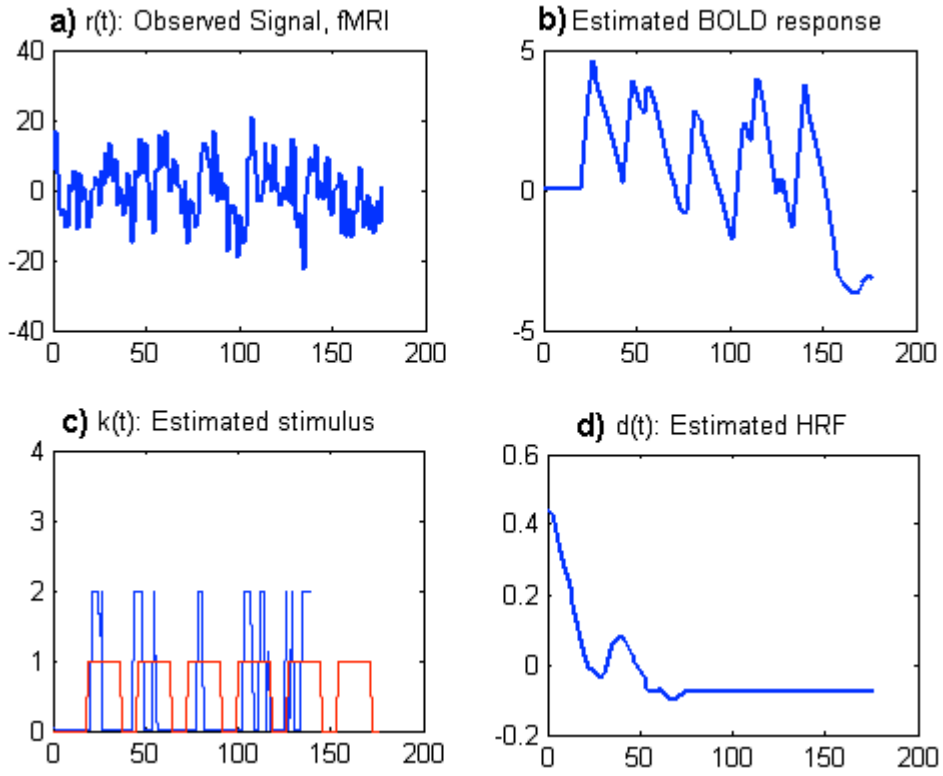


Figure 4.9 Estimated HRF, estimated stimulus and their convolution, "Estimated BOLD Response", given the real fMRI signal with ID_70

First of all, although the estimation of stimulus pattern (Figure 4.9c) in this case is not as good as the one in our simulation data, it is still good enough to locate the impulses only within the true impulse blocks. Moreover, the estimated BOLD response (Figure 4.9b) clearly has all peaks corresponding to the stimulus blocks except the last one. This is basically because in the observed fMRI signal, the last stimulus is not distinguished or visible, so the algorithm does not detect it. As for the estimated HRF signal (Figure 4.9d), we can argue that it is similar to our findings in the simulations except that the initial peak is followed by a smaller next one because of the categorical stimulus design. That is to say, the first peak in fMRI corresponding to the stimulus blocks is generated by initial peak of the HRF, and the second one of fMRI's is then generated by the smaller peak of estimated HRF. Normally, since we want to capture the impulses as our convolution filter, we upperbound the filter taps by 1. Indeed, the responses for each of

the category in blocks are expected to differ especially in the intensity point of view. But, for the sake of generalization, usually the applied stimulus blocks have the same intensity, which in turn forces to give the same impact. In reality, this is not the case. Instead when the category changes naturally its impact on the neural activity should also change. So, in this experiment which has stimulus blocks in different categories, the stimulus pattern should better be more relaxed. For that reason we may remove the upper bound on the convolution filter taps.

Next we study the fMRI signal from voxel with ID_100 in Figure 4.10. This voxel is extracted from our real data set with no upper bound convolution filter:

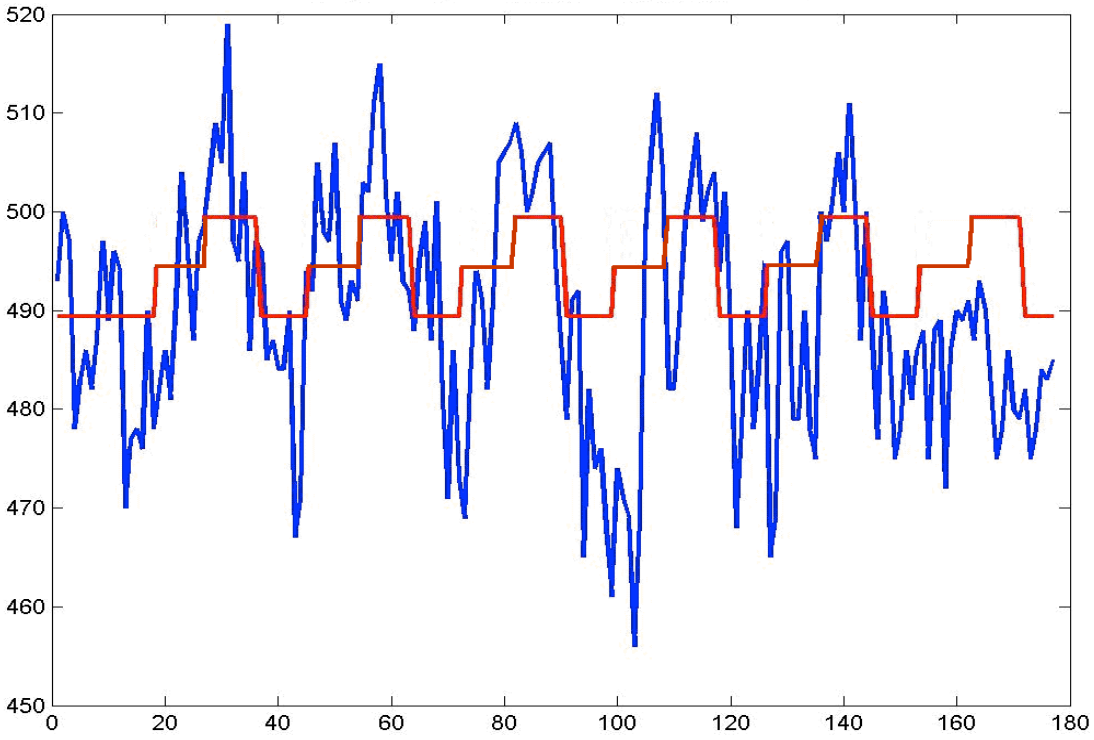


Figure 4.10 Real fMRI data with ID_100 and its stimulus pattern

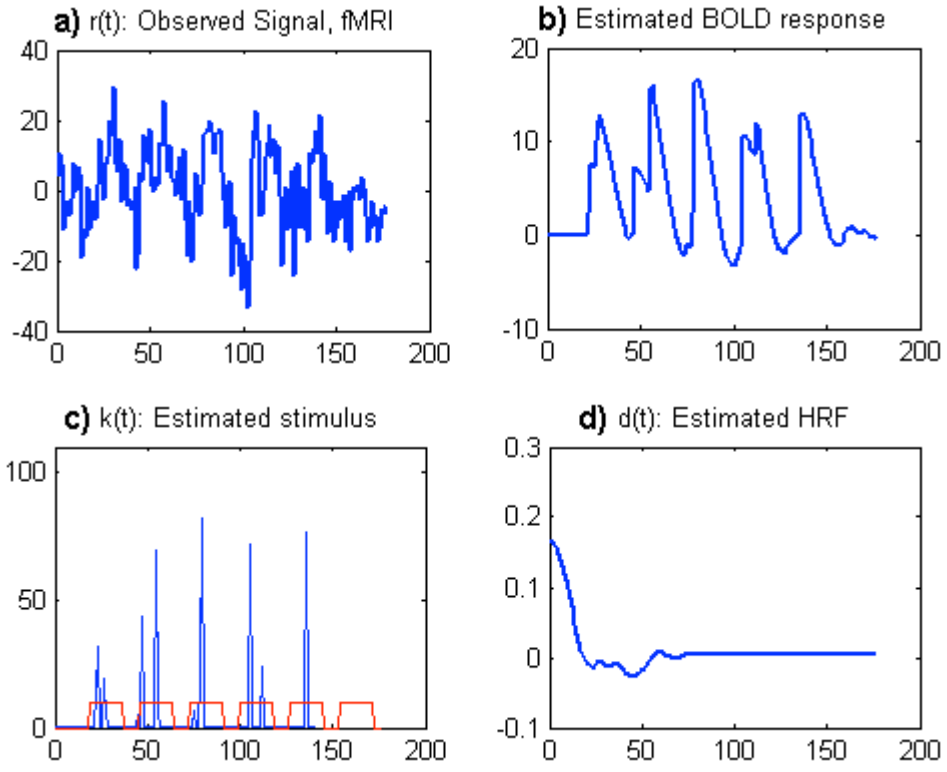


Figure 4.11 Estimated HRF, estimated stimulus and their convolution, "Estimated BOLD Response", given real fMRI signal with ID_100

The very first observation is that by removing the upper bound on convolution filter taps; we now are also able to detect the strength of the stimulus blocks (Figure 4.11c). For instance, in this example the strongest stimulus block is the one in the middle. Moreover, in the stimulus blocks, we detect two impulses, one is smaller and the other one is the larger. These impulses basically represent each categorical impulse in the underlying neural task, or represent the two successive peaks for each block in the fMRI signal. With the help of the two-peak detection for every block, we basically get rid of the second small peak of HRF of the previous case and obtain a better HRF estimation (Figure 4.11d). Similar to the previous case, there is no impulse detected in the last stimulus block (Figure 4.11b) since it is not distinguishable in the given signal, fMRI.

In the following, we show an example from voxel ID_215 from the class of passive fMRI time series, which represents the case when the voxel does not respond to the given stimuli. Figure 4.12 is the output of our algorithm:

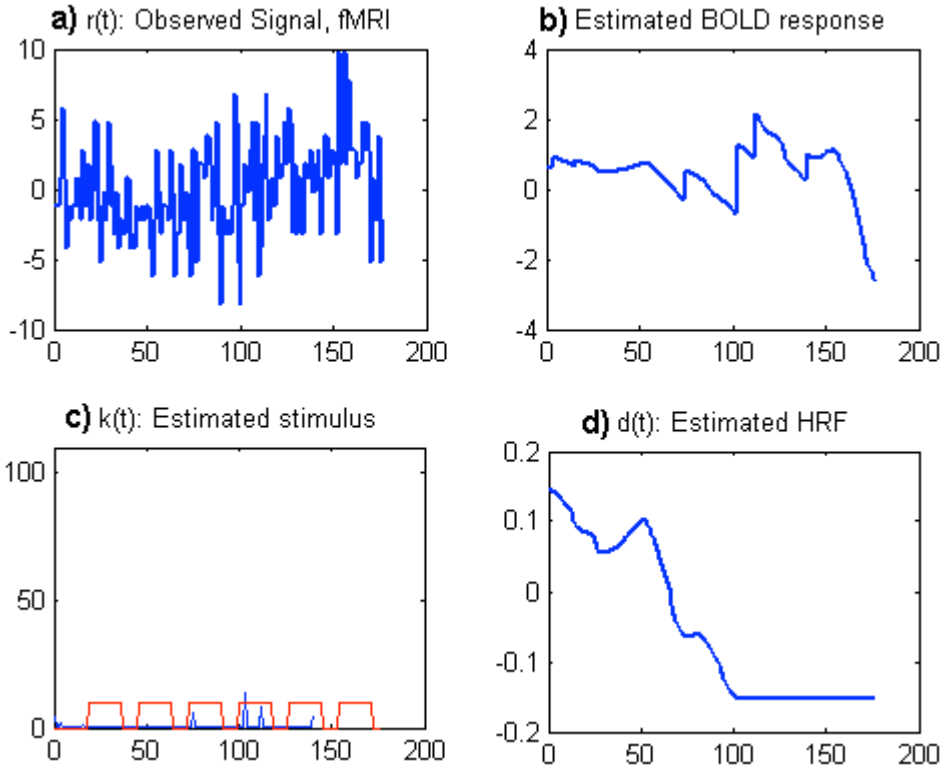


Figure 4.12 Estimated HRF, estimated stimulus and their convolution, "Estimated BOLD Response", given real fMRI signal with ID_215

As expected, this time the detected peaks in our estimated stimulus (Figure 4.12c) are very small ones and they are probably due to some intrinsic noise in the brain. Estimated BOLD response (Figure 4.12b) is hardly showing any sign for activation, mostly fluctuating around zero. The estimated HRF (Figure 4.12d) is the signal which is going to produce the observed fMRI through the convolution of estimated BOLD signal with very small impulses located in the middle. Hence, it spreads to a larger area in time compared to any other characteristic HRF.

At last, we conduct an HRF estimation experiment on the data set *'Data27'* which is from a finger tapping neural task that has a block stimulus as in Figure 4.13:

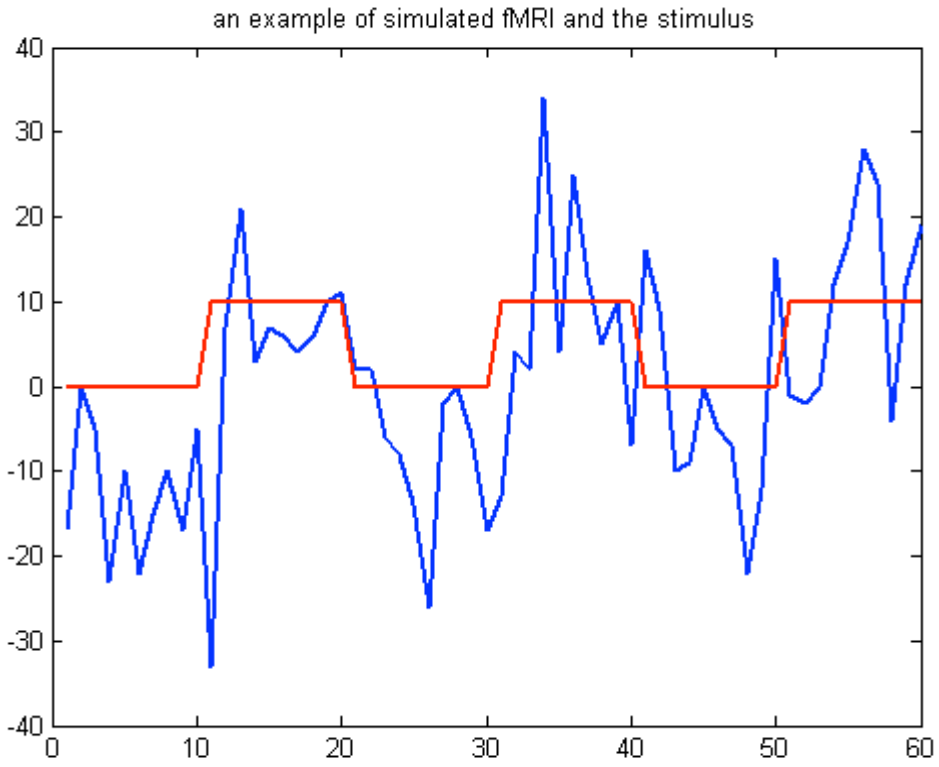


Figure 4.13 An example of real fMRI data from "*Data27*" with ID_11

In this experiment, once again the brain generates a valid physiological response to the motor stimulus with some delay of 2-3 time units. Our algorithm generates the results in Figure 4.14 for this example:

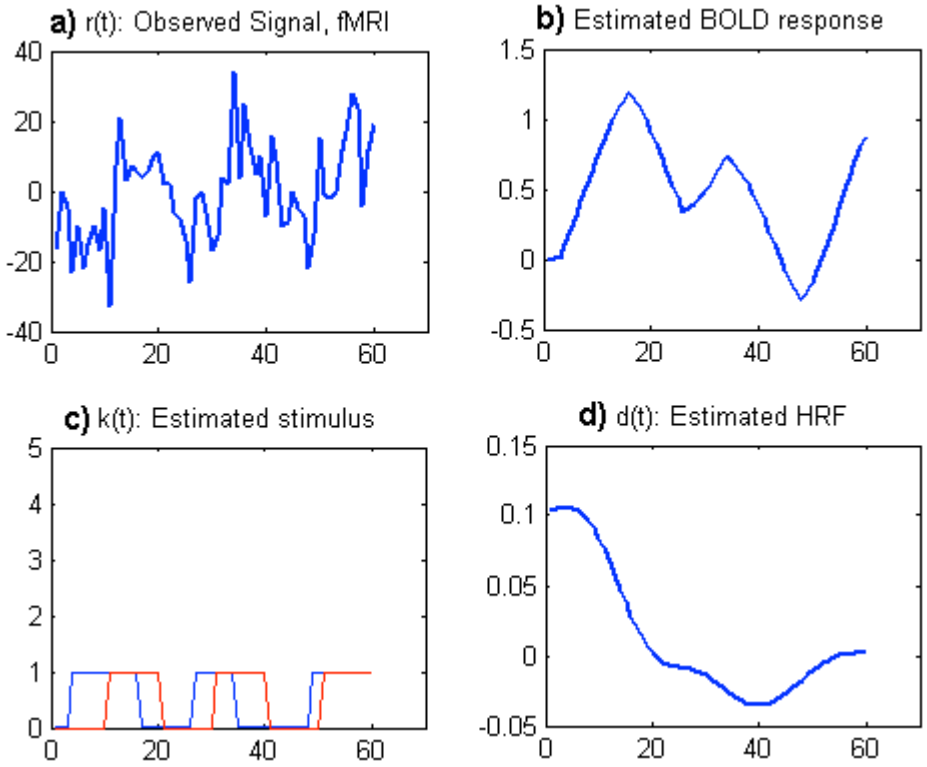


Figure 4.14 Estimated HRF, estimated stimulus and their convolution, "Estimated BOLD Response", given real fMRI signal from "Data27"with ID_11

The true stimulus pattern is estimated with a correct amount of duration and periodicity but it is shifted (Figure 4.14c). Ideally the estimated HRF (Figure 4.14d) should have been narrower with a quicker decrease after hitting the peak. However, our smoothness constrain forces it to be a little wider and for that reason it creates a larger delay. As a result, to compensate the large delay that the HRF puts, the estimated impulse pattern begins a little earlier.

Modification for HRF Extraction

To sum up, the stimulus patterns estimated by our algorithm are acceptable. As for the estimated HRFs, in order for it to settle, κ is chosen to be small. But then we miss the initial rise of the ideal HRF. To adjust for the initial rise, κ should be increased, but then HRF does not settle and starts fluctuating. This is because we use the same penalties for derivatives at every time point. However, ideally HRF has large derivative values in the beginning and small ones in the end. To overcome this issue, and also to improve the overall HRF estimation, we adjust the smoothness constraint accordingly:
Recall that the smoothness constraint was:

$$\begin{aligned} \arg\min_{\mathbf{d}} smoothness \quad (\mathbf{d}) &= \arg\min_{\mathbf{d}} \sum_{i=0}^{N-2} (d(i)-d(i+1))^2 + (d(N-1)-d(N-2))^2 \\ &= \arg\min_{\mathbf{d}} \mathbf{d}^T\mathbf{L}^T\mathbf{L}\mathbf{d} = \arg\max C_d\, e^{\frac{-\mathbf{d}^T\mathbf{L}^T\mathbf{L}\mathbf{d}}{2\lambda^2}} \end{aligned} \tag{4.2}$$

Instead of it, we use time dependent derivative penalties, meaning that at every time instant we apply a different penalty for the derivative:

$$\begin{aligned} \arg\min_{\mathbf{d}} smoothness \quad (\mathbf{d}) &= \arg\min_{\mathbf{d}} \sum_{i=0}^{N-2} \mathbf{Z}_i^2 (d(i)-d(i+1))^2 + \mathbf{Z}_{N-1}^2 (d(N-1)-d(N-2))^2 \\ &= \arg\min_{\mathbf{d}} \mathbf{d}^T\mathbf{L}^T\mathbf{Z}^T\mathbf{Z}\mathbf{L}\mathbf{d} = \arg\max C_d\, e^{\frac{-\mathbf{d}^T\mathbf{L}^T\mathbf{Z}^T\mathbf{Z}\mathbf{L}\mathbf{d}}{2\lambda^2}} \end{aligned} \tag{4.3}$$

where **Z** is a diagonal penalty matrix. Ideally, **Z** values for the first time instants should be smaller than the rest, so that we allow larger variations in the beginning of estimated HRF than the ones in the end.

Originally, we assign uniform penalty to large derivatives throughout the estimated HRF. We adjust it by incorporating the diagonal matrix **Z** in our MAP estimation, such

that instead of minimizing $|\mathbf{Ld}|^2$, we minimize $|\mathbf{ZLd}|^2$ where the diagonal entries of $\mathbf{Z}$ keep the penalty values for a unit derivative at every time instant. Hence, if we put large values in the first diagonal entries and small values in the rest of the entries, we get the desired effect. Mathematically, this results in a prior Gaussian distribution on the HRF which does not have a circular shape but an elliptic one. This kind of change can easily be incorporated within our Bayesian approach.

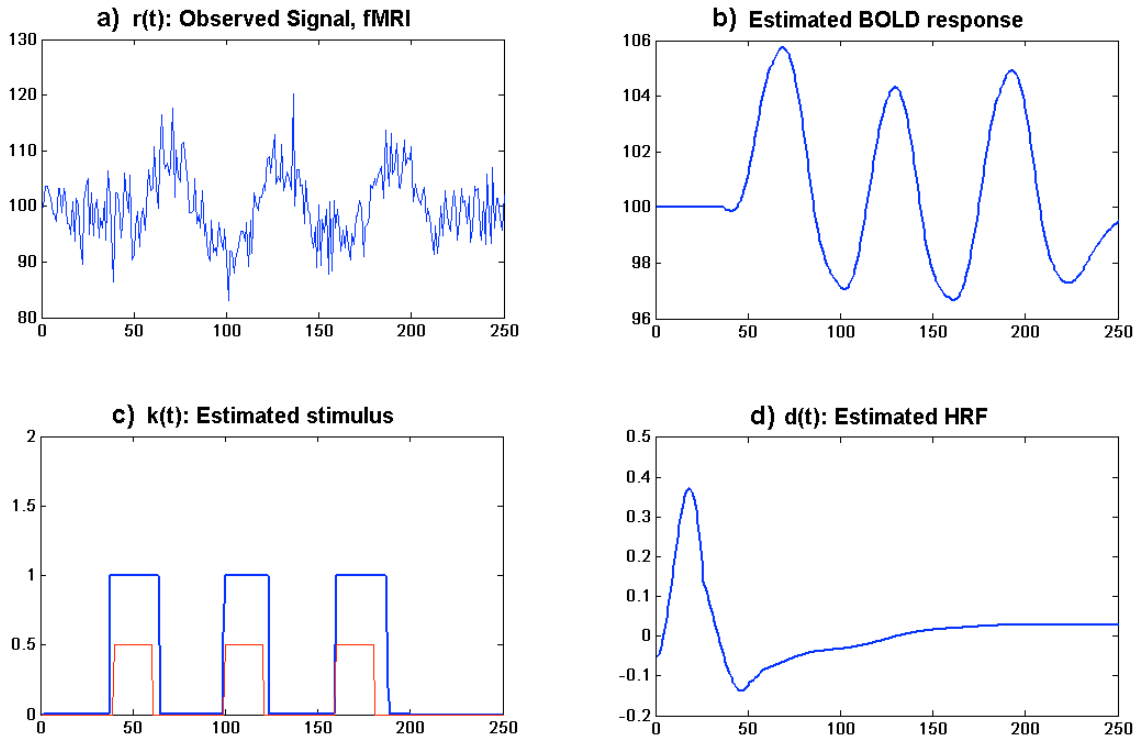


Figure 4.15 Modified HRF of the simulation data with σ_{AWGN}=4

With the help of this modification, the estimated HRF of the previous simulation data with σ_{AWGN}=4 as shown in Figure 4.6d becomes closer to the ideal HRF and we can obtain the initial rise of the HRF as it is seen in Figure 4.15d. Also, the estimated stimulus pattern (blue one) in Figure 4.15c almost matches the unknown true stimulus (red one).

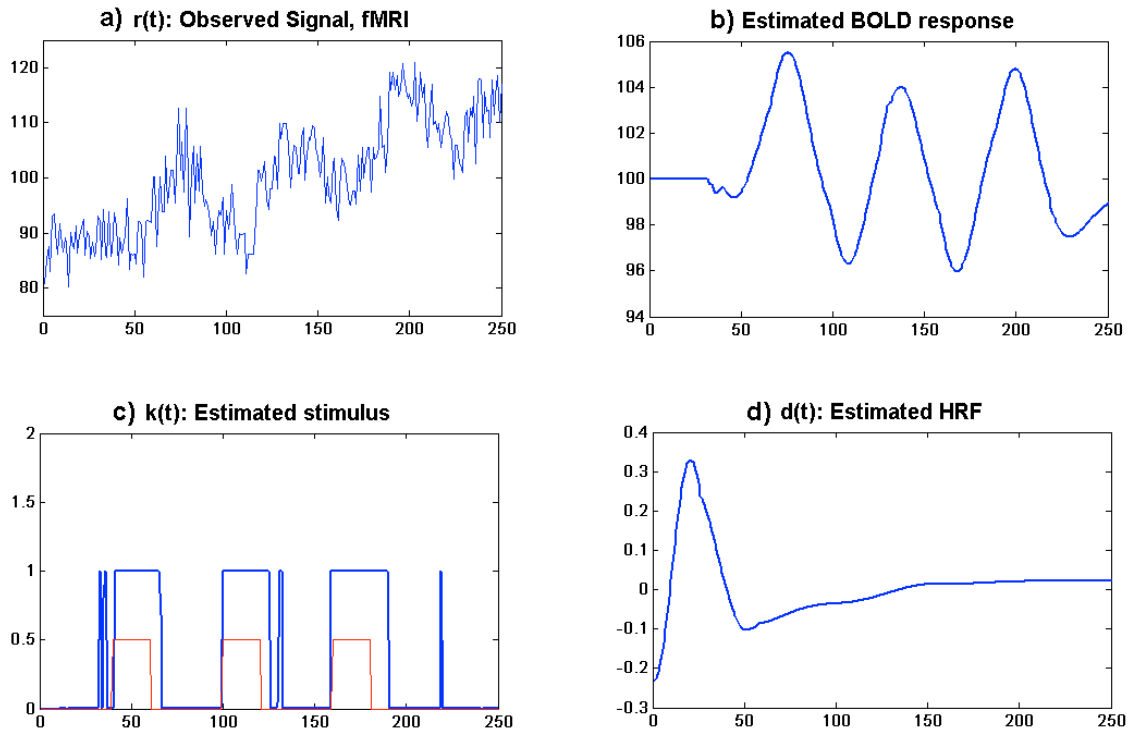


Figure 4.16 Modified HRF of the simulation data with σ_{AWGN}=4, σ_{jitter}=4, σ_{lag}=16, σ_{drift}=16

Even if we have a noisier fMRI signal with σ_{AWGN}=4, σ_{jitter}=4, σ_{lag}=16, σ_{drift}=16 (Figure 4.16a), our algorithm can still estimate HRF with initial rise as shown in Figure 4.16d. Stimulus pattern is also estimated well except that a few incorrect impulses are observed in the estimated stimulus as in Figure 4.16c.

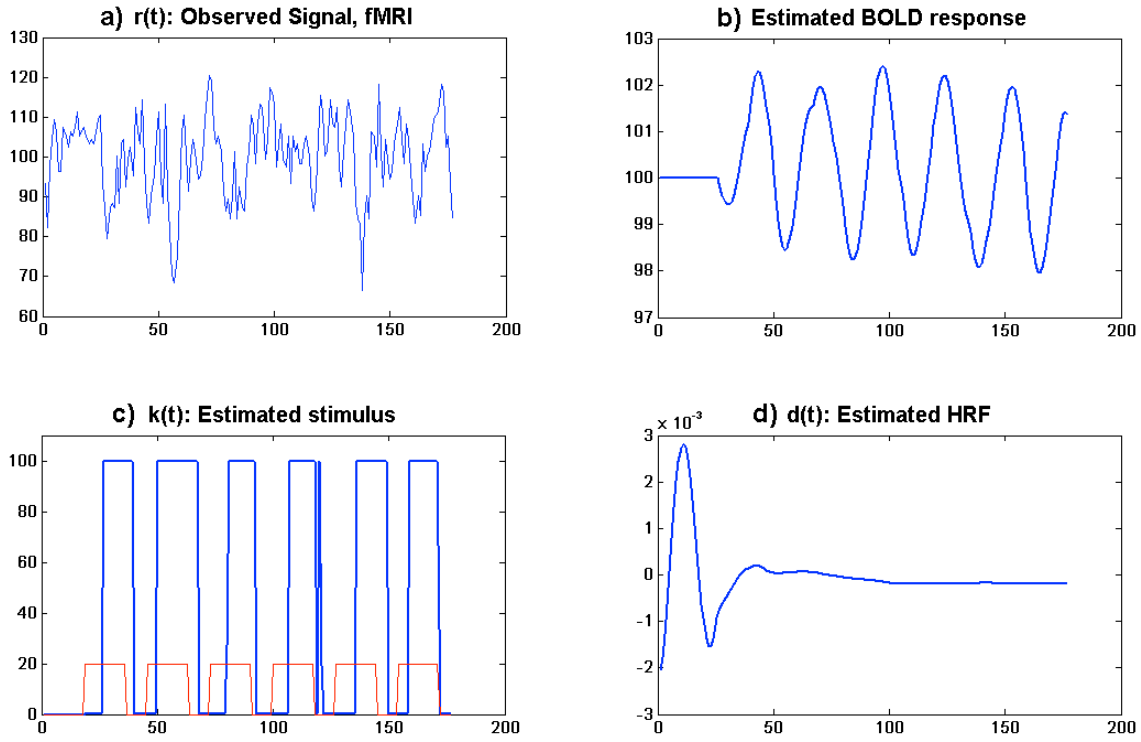


Figure 4.17 Modified HRF of the real data with ID_6

Our real fMRI experiments produce results as good as the ones in simulations as it is seen in Figure 4.17 (real fMRI data from the experiment with successive categories). If we take a closer look on the estimated HRF (Figure 4.17d), we notice that it is very close to the ideal shape of HRF.

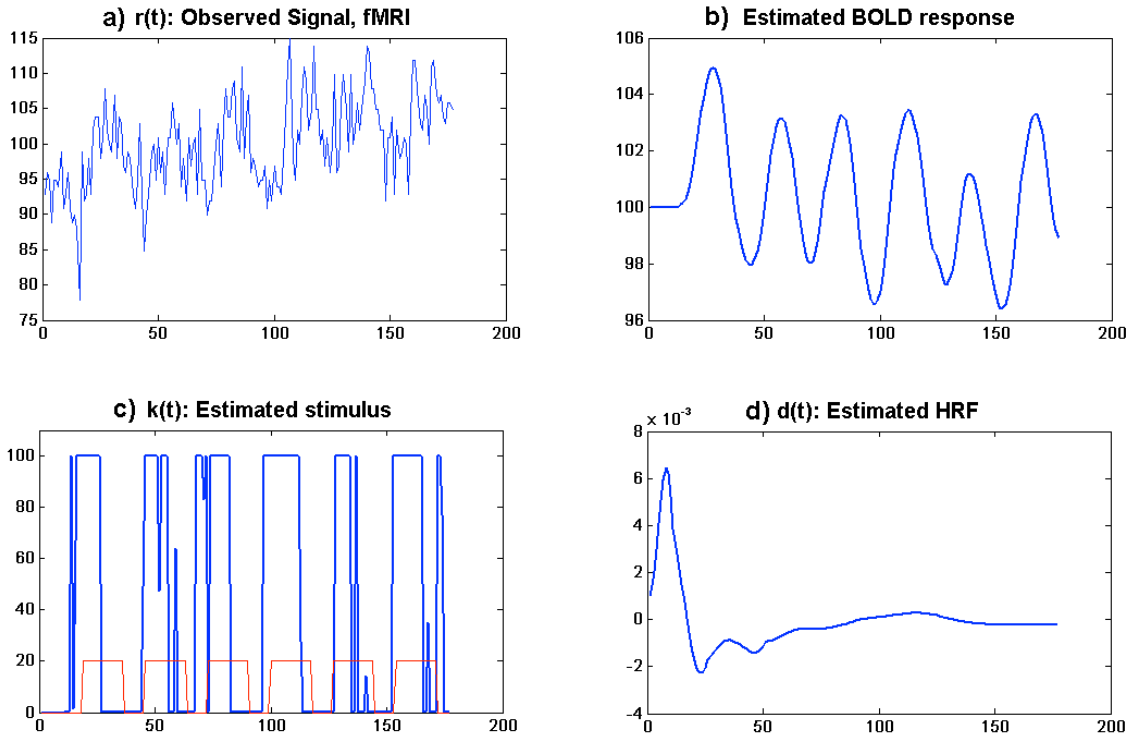


Figure 4.18 Modified HRF of the real data with ID_135

Figure 4.18 shows another example voxel from the same real fMRI dataset. Here the HRF is estimated satisfactorily (Figure 4.18d). As for the estimated stimulus pattern (Figure 4.18c), our findings show that, the external stimulus does not truly show itself in the fMRI signal and the subject cannot perceive the whole applied stimuli. For example, the subject may not perceive the entire stimuli or may not attend similarly throughout the experiment. This explains why we do not prefer to use external applied stimulus as input in our approach.

Estimated hemodynamic response functions give the characteristics of the corresponding voxel. They can be used as input to clustering in order for determining functional similarities of voxels. But due to the variability of the HRFs we preferred to use hemodynamical time series in clustering analysis. In the following section we present the estimated hemodynamical time series and the advantages of using them as input for our spectral clustering algorithm.

Hemodynamical Time Series

Hemodynamic response is essential for a better understanding of the activation characteristics of the related voxel. But, for our clustering approach using HRFs of voxels does not yield satisfactory results since HRF is intrinsic to the neural tissue underlying a specific voxel or a group of voxels and it is not the same for all active voxels. That means, even if the general shape of the active HRFs are similar, their intensity, rise time, durations and the fall time can vary from voxel to voxel. Due to the variety of HRFs, using a single HRF for representing the related voxel is not convenient for grouping the voxels according to their activation. Instead, in this thesis, we use "hemodynamical time series" as input to our clustering algorithm.

Hemodynamical time series includes the characteristics of HRF and also the information of the underlying stimulus such as the location of stimulus blocks as well as the duration of them. In some way, it represents a noise-free physiological response for the stimulus perceived by the subject. Throughout the experiments, at some period, the subject may not concentrate on the task; therefore the activity in the responding voxels may slip. This also affects the obtained HRF, so the clustering algorithms using pure HRFs may have misleading results. In another example, the subject may move his head at some instant yielding a defected HRF derived from this interference. But, these will affect the response of all voxels at the same instant with same durations. For instance, active voxels will be inactivated or less activated during the same time period, causing an unexpected variation in the fMRI signal. However, this variation will be consistent throughout the voxels that are similar in their functionality, allowing for their clustering together. Unfortunately, in the extracted HRFs, we cannot observe properly these defects, more importantly we cannot know the timing of the actual stimulus. Since hemodynamical time series contain response to the stimulus perceived by the subject on the spot, we can clearly realize the activation due to the actual stimulation for all voxels. As a result our clustering algorithm will be immune to this type of problem.

In the following sections we will give some examples on hemodynamical time series of real fMRI data sets. The example signals are chosen to clearly show that the evoked neural activity is not always the perfect response to the applied stimulus rather it is the response only to the actually received underlying stimulus. As we claim that the receipt of actual stimulus may differ from voxel to voxel, then it only natural that the related neural responses differ as well. Here, the effect of the actual stimulation on the neural response will be clarified making the reason for using the hemodynamical time series more evident. Also, we will put forward some physiological inferences and comments on the neural responses.

4.1.1 Block Design Hemodynamical Time Series

In the block design experiment, fMR images are obtained, in a classical finger tapping paradigm, as 60 time points are collected in 3 cycles which contained 10 samples for each ON or OFF periods through the echo planar imaging protocol.

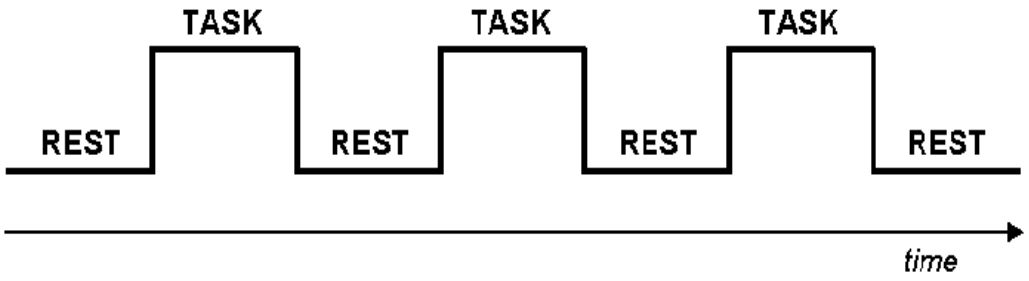


Figure 4.19 fMRI Block Design Paradigm

Active and passive voxels are classified beforehand via general linear model, which served as 'ground truth'.

Below, the fMRI data generated using block design stimuli and the corresponding hemodynamical time series are shown:

a) Active Voxels:

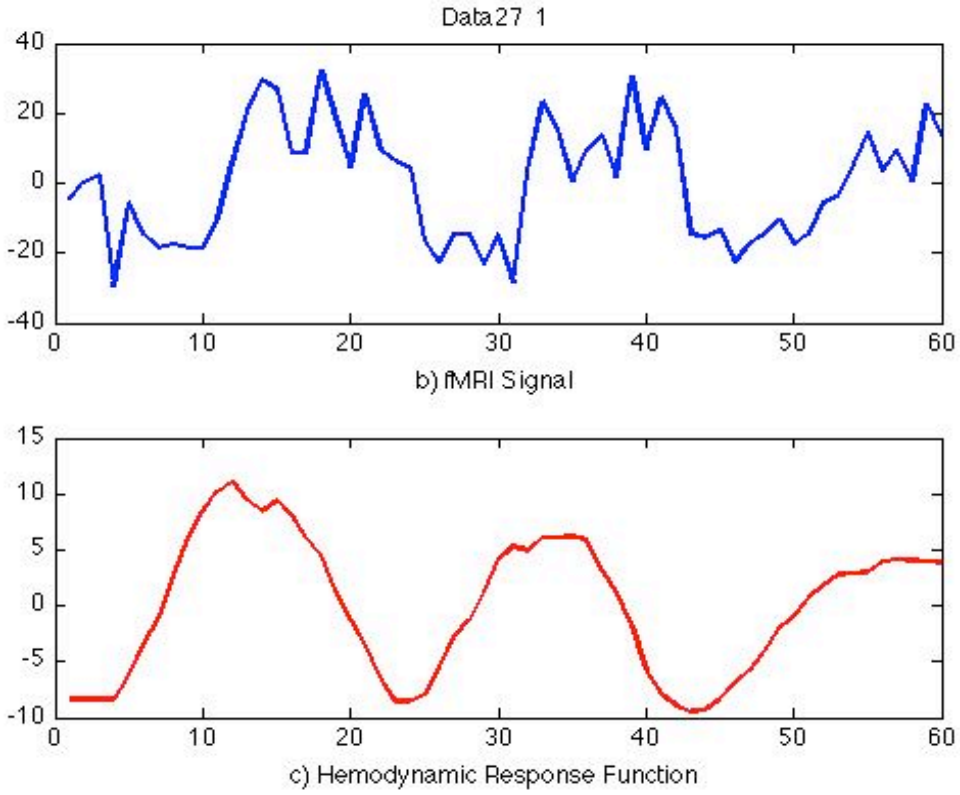


Figure 4.20 a) fMRI Signal b) Hemodynamical Time Series obtained from 1st voxel of Data27

If we present a categorically similar stimuli block to the subject, we obtain activations as in Figure 4.20. If only one stimulus is delivered, a small peak would be observed as activation, which sometimes may be too small to be detected. However as the number of the applied stimuli is increased to form a block, the hemodynamical activations will exhibit a cumulative effect through convolution. Since the type of stimulus is fixed within a block, the evoked brain activity is thought to be the same for each stimulus. However, this assumption fails to consider changes over time in arousal and attention. Even when fully alert, subjects are often thinking about something other than the experimental task. Moreover, if one repeatedly presents the same stimulus blocks to a

subject, for example when giving a mild electric shock to a rat or showing a bright red balloon to a human infant, the response to that stimulus will diminish over time. This explains why the amplitude of the second peak is smaller than the first one.

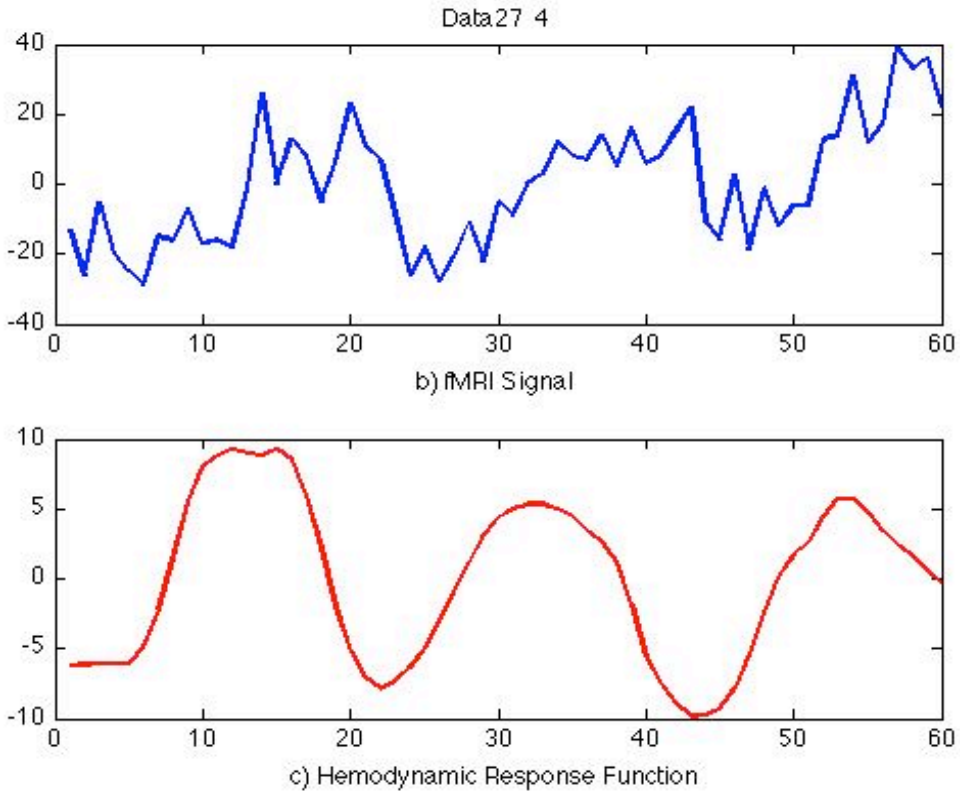


Figure 4.21 a) fMRI Signal b) Hemodynamical Time Series obtained from 4th voxel of Data27

The voxel we consider in Figure 4.21 is active according to the applied blind deconvolution method. There are three precise neural peaks as in Figure 4.20. But the intensity levels in Figure 4.21 are less than that of Figure 4.20. Even though the subject sees the same stimulus and performs the same action in each trial, the amplitudes of these activations are smaller. The presented experimental tasks evoke activity in a set of related brain regions. Since the task is finger tapping, it will elicit activity in motor

regions of the brain such as primary motor cortex, supplementary motor cortex, and cerebellum. But all motor regions will not be activated at the same level, which will yield differences in the activation levels as in Figure 4.20 and Figure 4.21.

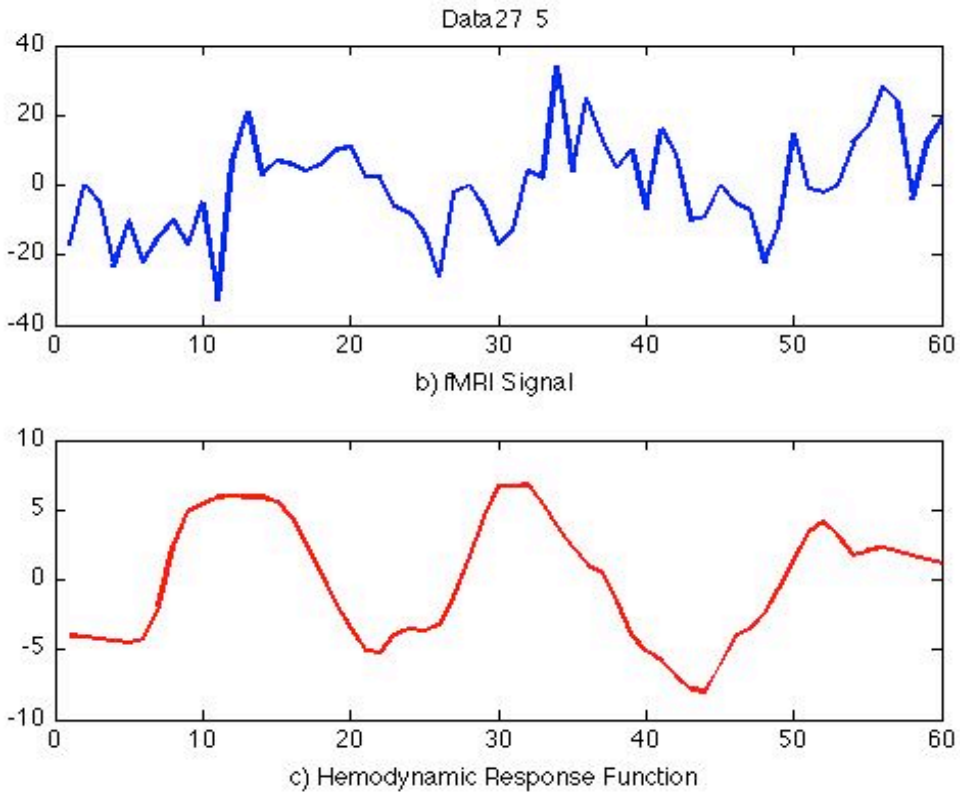


Figure 4.22 a) fMRI Signal b) Hemodynamical Time Series obtained from 5th voxel of Data27

The obtained hemodynamical time series in Figure 4.22 indicates that this voxel also becomes active when faced with this type of stimuli. When the stimulus is presented in blocks with high frequencies, the activation peak extends into a plateau as in the first activation period in Figure 4.22. Here, in the second activation period, the cumulative activation stands for a shorter period. It is expected that it would form a plateau and stay in saturation until around the time point 40 as in Figure 4.20 and Figure 4.21. But, the intensity decreases after a few time points period. During the experiments, subjects may

lose their attention as they are hearing sounds of the scanner gradients, receiving varying visual stimuli as they look around and may be prone to the mental imaginary activities. Also subjects may get tired over time and their performance may worsen as the experiment goes on. As a result even if the applied stimulus has 10 time points of duration, the actual stimulus that evokes the neural activation seems to have smaller duration. This can explain the shortness of the activation interval and the early decay of the second activation in Figure 4.22.

b) Passive Voxels:

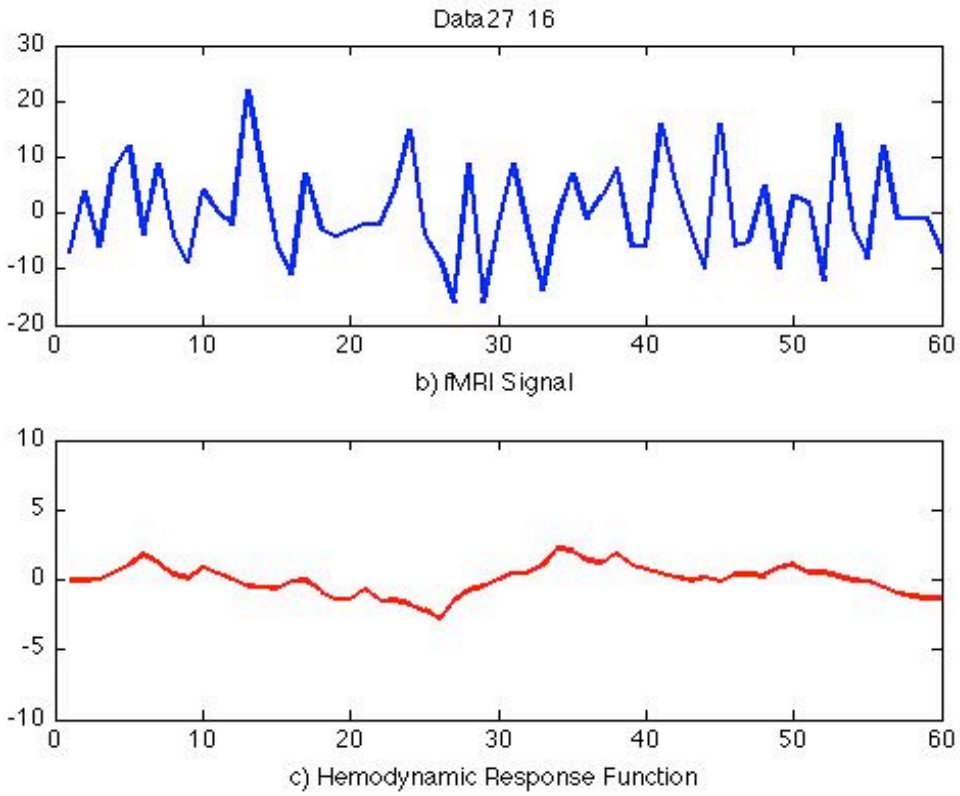


Figure 4.23 a) fMRI Signal b) Hemodynamical Time Series obtained from 16th voxel of Data27

In the passive voxels, the fMRI signal has no meaningful profile. These changes emerge from temperature fluctuations in the scanner or the subject's body and physiological effects like head motion, heart rate and respiration, which occur together with an experimental manipulation. Although these artifacts are diminished during data preprocessing, they cannot be totally discarded. Furthermore non-task related neural variability cannot be totally eliminated. During the finger tapping experiment, the feeling evoked on the subject's finger may become an irrelevant stimulus for the neural tissue in some voxels. Figure 4.23 shows such a routine neural variability.

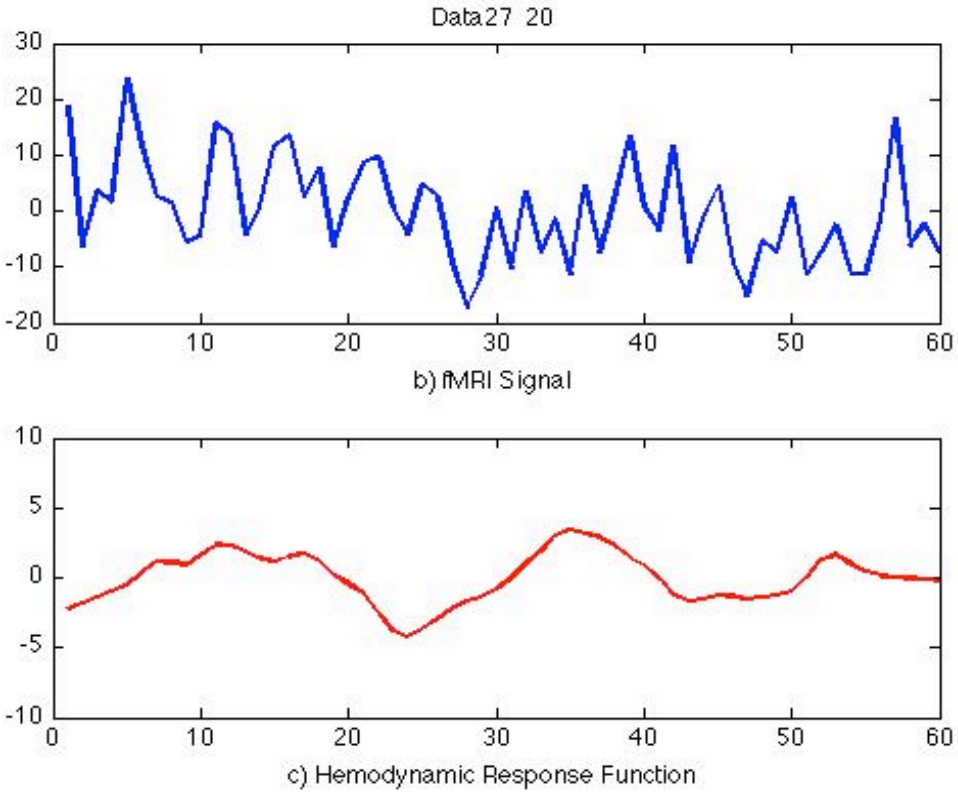


Figure 4.24 a) fMRI Signal b) Hemodynamical Time Series obtained from 20th voxel of Data27

The obtained hemodynamic time series in Figure 4.24, belongs to a passive voxel as in Figure 4.23, since no Gaussian-like peaks are observed. The signal intensity increases at some time points, but is not enough to decide that this voxel is activated, because the rises are independent from the applied stimulus.

Similarly, the variation in the signal level of the obtained hemodynamical time series does not alter considerably during the experiment time in Figure 4.25. Therefore, this signal also comes from a passive voxel.

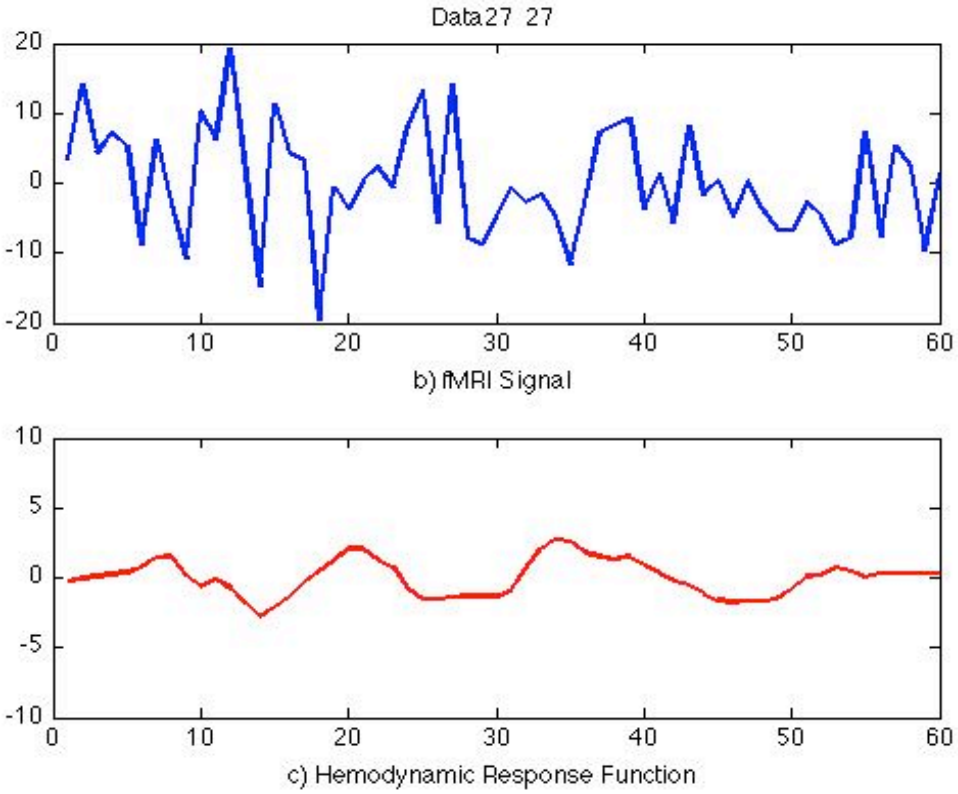


Figure 4.25 a) fMRI Signal b) Hemodynamical Time Series obtained from 27th voxel of Data27

4.1.2 Categorical Block Design Hemodynamical Time Series

After obtaining reasonable results using block design experiments and in order to better validate the followed methodology, new and different types of fMRI data are used. Here the applied stimuli are again in block but 2 categories exist within a block.

In this categorical block design experiment, fMRI data is conducted in a 1.5T Siemens scanner. It is an fMR adaptation paradigm consisting of 177 time samples, investigating subtle effects in face processing.

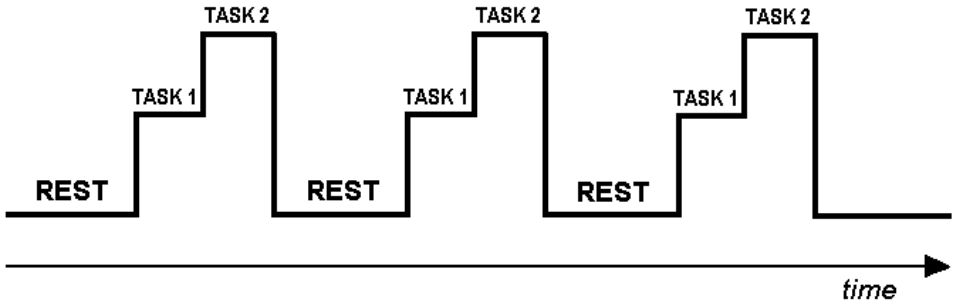


Figure 4.26 fMRI Categorical Block Design Paradigm

If face recognition task is represented as 1, rests can be represented simply as 0. On a period of nine time points, faces in one category are shown, after that other category of faces are shown on a period of nine time points. After these two tasks are performed a period of nine points of rest is given, that means no face pictures are shown. This is an experiment with 6 cycles. The experiment begins with eighteen time points of rest. Then one period of stimulus series will be basically 111111111 111111111 000000000.

Task: Block paradigm, face perception: 177 sample points:

9 dummies at the beginning (0)

9 patches (0) ----------

9 faces (1) ----------

9 faces (2) ----------

(this group of 27 samples is repeated 6 times)

6 dummies at the end

In the following, the fMRI data generated using categorical block design stimuli and the corresponding hemodynamic responses are shown:

a) Active Voxels:

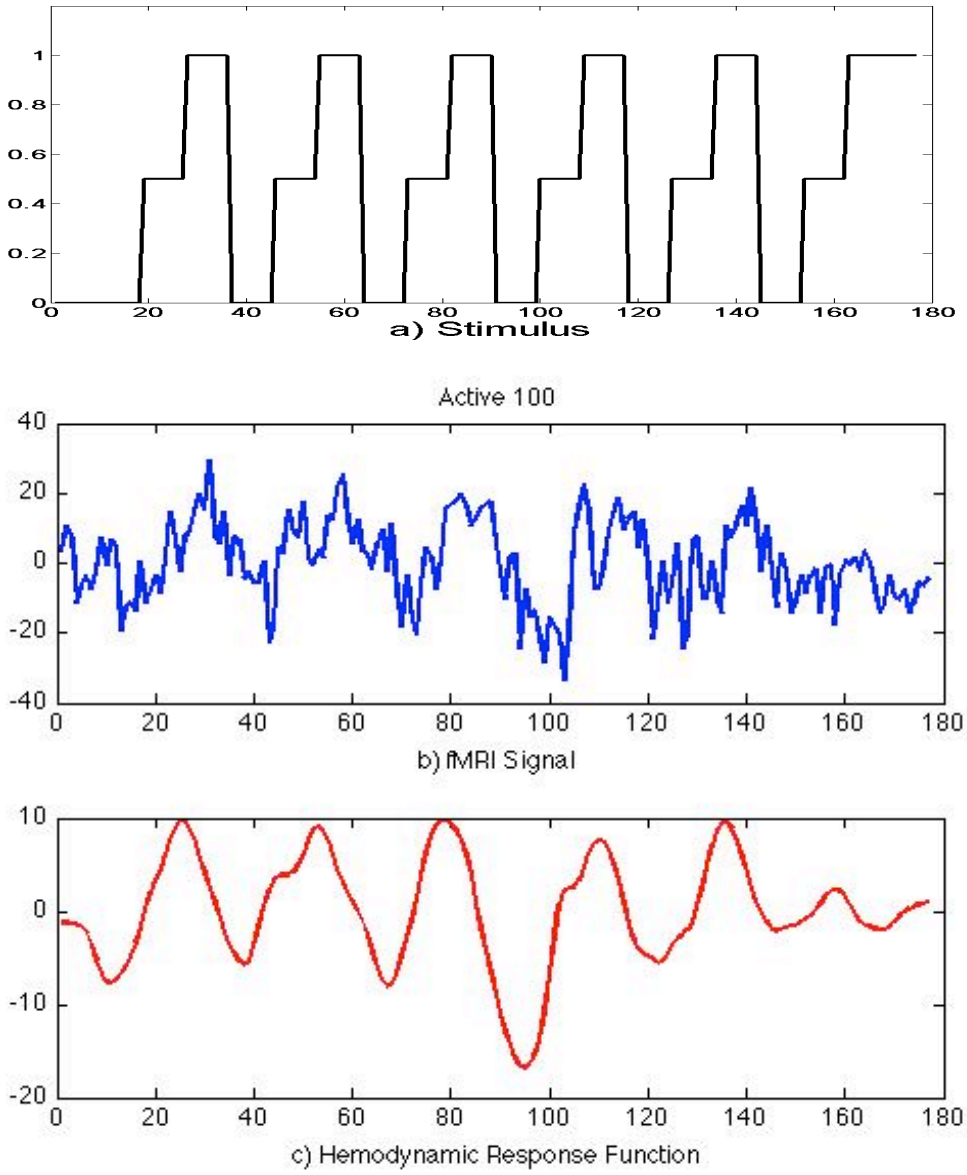


Figure 4.27 a) Applied stimuli b) fMRI Signal c) Hemodynamical Time Series obtained from 100th voxel of Active_male_1

During the repeated stimuli representations, the response to that stimulus will diminish over time which is due to the nonlinear characteristics of the fMRI and is known as refractory effects [68]. When the response to a stimulus decreases with repeated presentation, the voxel is said to have adapted to that stimulus. By changing the stimulus slightly, one can infer to which aspects of the stimulus the voxel adapted. Many voxels in the visual regions of the brain will preferentially fire for one type of visual stimulus, due to its particular shape, color, contrast, or motion. But with repeated presentations of that stimulus, the voxel's firing rate will decrease.

In the face perception experiment, face images that evoke activity in the visual regions of brain are presented repeatedly, After 9 time points repetition, than another type of faces that differ in some fashion are presented. If the voxels being investigated respond differently to the new face stimulus, it will show an increase in fMRI activity. But if the voxels do not distinguish between the old and new stimuli, there would be no increase in the fMRI intensity level compared to the old stimuli presentation. In Figure 4.27, as stimuli are presented in succession, each contributes to the total activation. With long task blocks, which are greater than the width of the hemodynamic response to a single stimulus, every time point within the block contains a contribution from multiple stimuli, each at different phase. Because of the refractory effects in the fMRI signal, if there were no change in the category of the face images, we would expect the activation increases rapidly at the onset of the task, thereafter remaining at the plateau value until the cassation of the block, or more dramatically it would start to fall before the second task block started. The activations in Figure 4.27 triggered due to the both categories of faces and the related voxel fire for the differences of the stimuli blocks presented in succession. If the change in the category of the stimuli would not be detected by this voxel, the intensity levels of the peaks will be smaller as well. However for the last task block there is no activation. That indicates there is no actually stimulated neural activity during this interval.

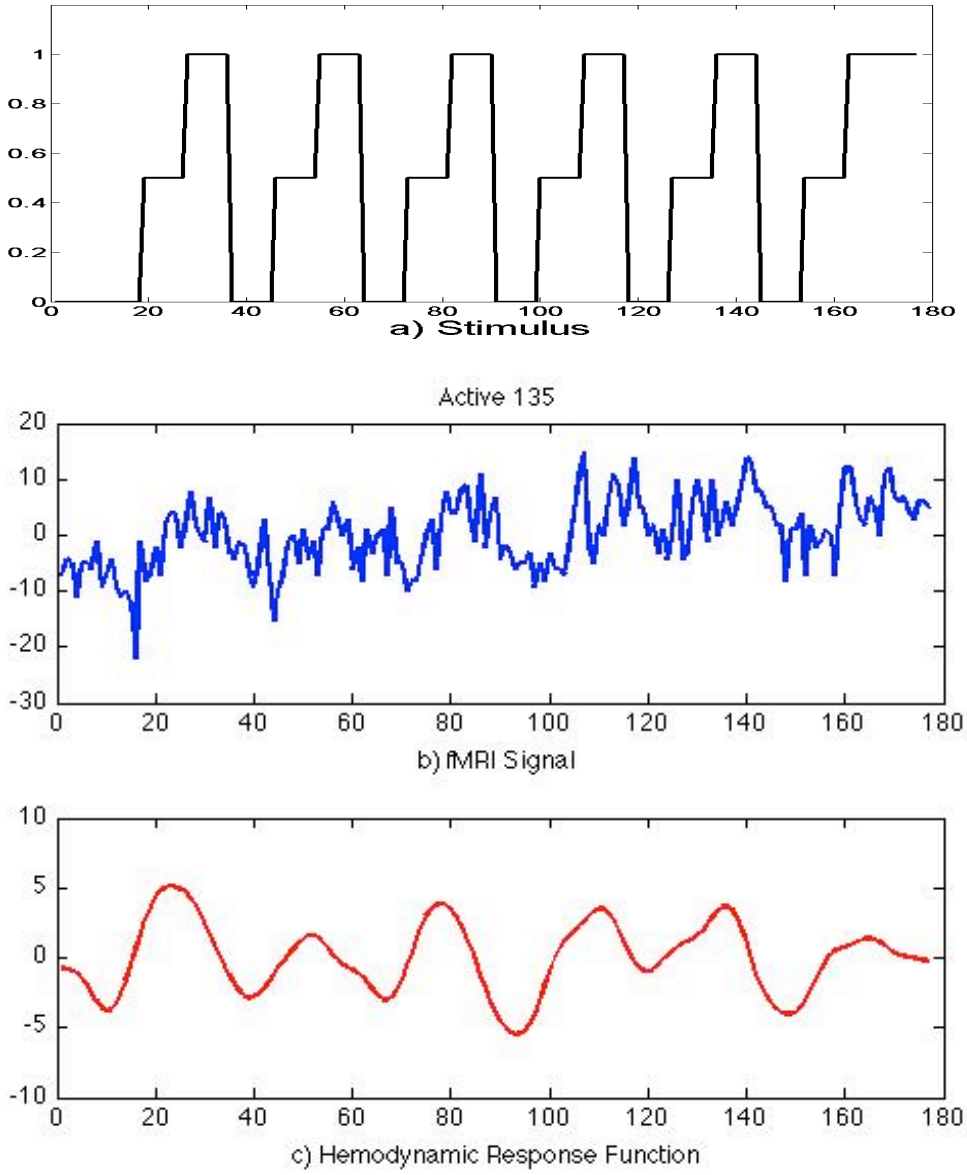


Figure 4.28 a) Applied stimuli b) fMRI Signal c) Hemodynamical Time Series obtained from 135th voxel of Active_male_1

A hemodynamic response with large-amplitude might reflect greater metabolic demand, and thus greater neural activity. Also, since veins have much greater volume than capillaries, there would be a greater change in the amount of the deoxygenated hemoglobin and thus in the total hemodynamic response within the voxels nearby the veins. That means, some brain regions have greater blood flow than others, and some regions have higher oxygen requirements than others independent from the task of being concerned. However, the basic assumption of all block design paradigms is that block related changes result from the differences between the experimental conditions that are task and control. The task condition is assumed to contain all of the neural processes present in the control condition as well as additional processes related to the task. Also, the intensity levels of the activations are to provide information about the relative difference between the two conditions, not about the absolute levels of activity. For the purpose of proper analysis, an appropriate baseline is crucial so that all experimental conditions can be compared relative to each other [69]. In Figure 4.28, even though, there are six clear activation peaks, it does not have a clear baseline but a quadratic drift in the raw fMRI data. We eliminated this drift and that is why the obtained hemodynamical time series has a settled baseline.

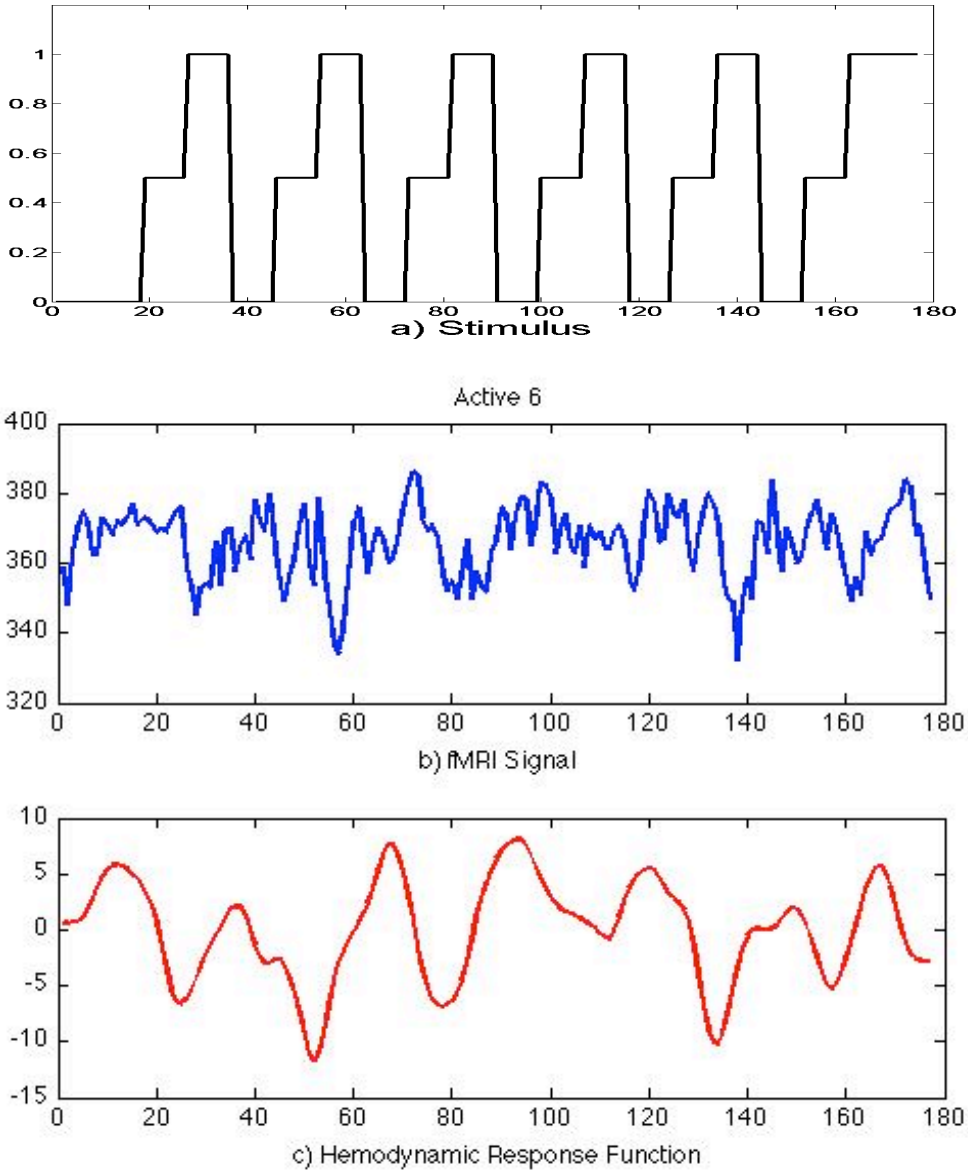


Figure 4.29 a) Applied stimuli b) fMRI Signal c) Hemodynamical Time Series obtained from 6th voxel of Active_male_1

The duration of the stimulus does not necessarily correspond to the duration of the neural activity. During the face presentation experiment, the voxels in the primary visual cortex of the subject exhibit their largest response immediately after the presentation of the first stimulus and then become much less active. Voxels within the face-sensitive regions in the inferior temporal lobe are active at stimulus onset, as well as throughout the stimuli block period due to the feedback from other brain regions. Some voxels within the frontal and parietal lobes may become active after the face disappears reflecting what the subject just saw. Moreover, when the stimulus is presented, a response time for subject is needed to perceive the face and thus yielding a delay for the activation. However, the activation corresponding to the onset of the stimulus may also exhibit an early response in time compared to those responses with offset to the stimulus administration time. This early response reflects the preparation process, due to the subject knowing or anticipating when the stimulus would appear. In Figure 4.29, first activation appears before the stimuli presentation.

b) Passive Voxels:

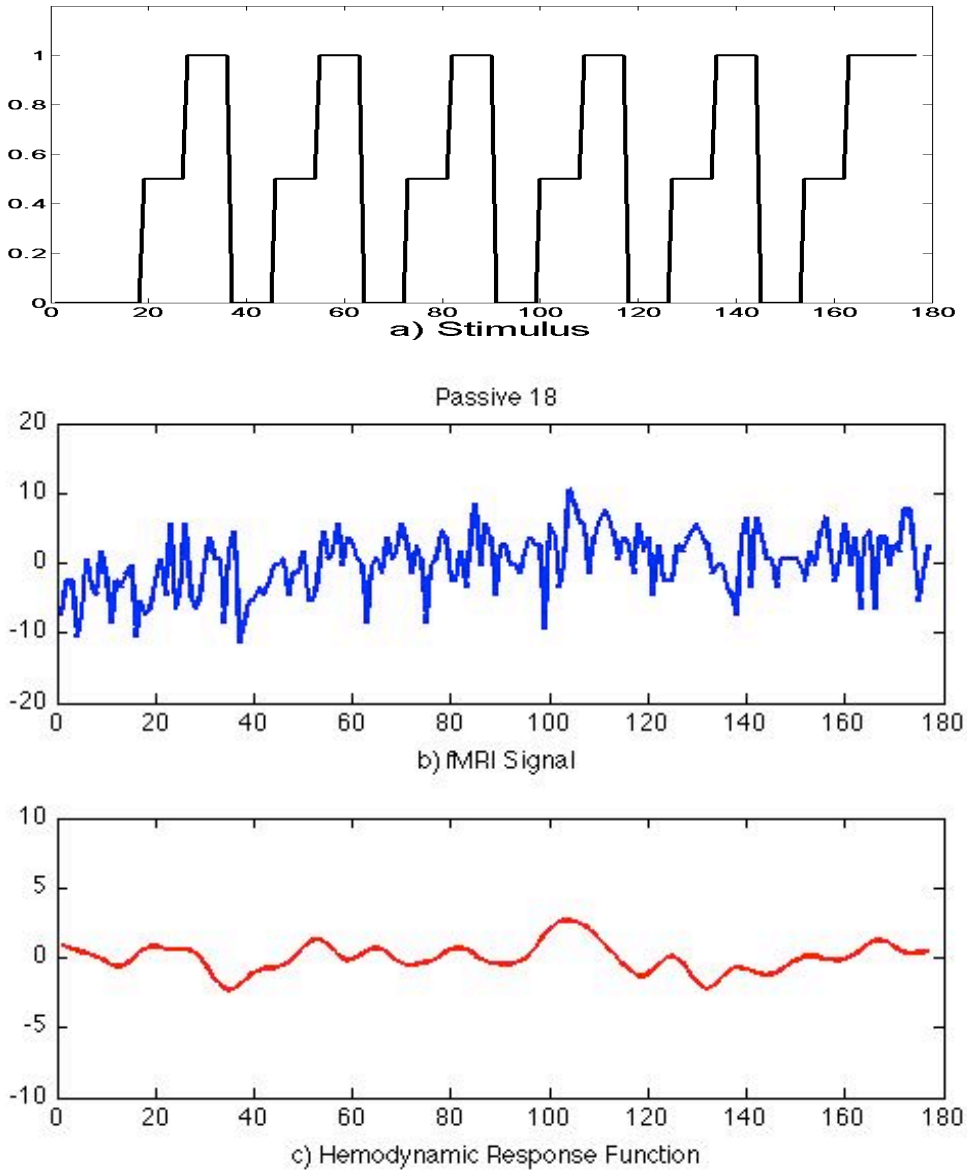


Figure 4.30 a) Applied stimuli b) fMRI Signal c) Hemodynamical Time Series obtained from 18th voxel of Passive_male_1

In the face perception experiment, generally we attempt to investigate which areas of brain are activated by visual stimuli. However, the subject hears the sounds from the surrounding, feels the cold within the scanner, and looks around during the time between the two successive stimuli, at the same time that he experiences the task-related stimulus. All of these internal and external stimuli yield activation in the brain and thus affect the voxel intensity over time. But the changes in the intensity level, as seen in the Figure 4.30, are random since they are unrelated to the stimulus of interest. These non-task-related stimuli randomly evoke neural activities in any area of brain, which emerge independently from the concerned stimuli timing. Since there are no systematic peaks due to the applied stimuli blocks, this voxel is not activated by the stimuli blocks of interest.

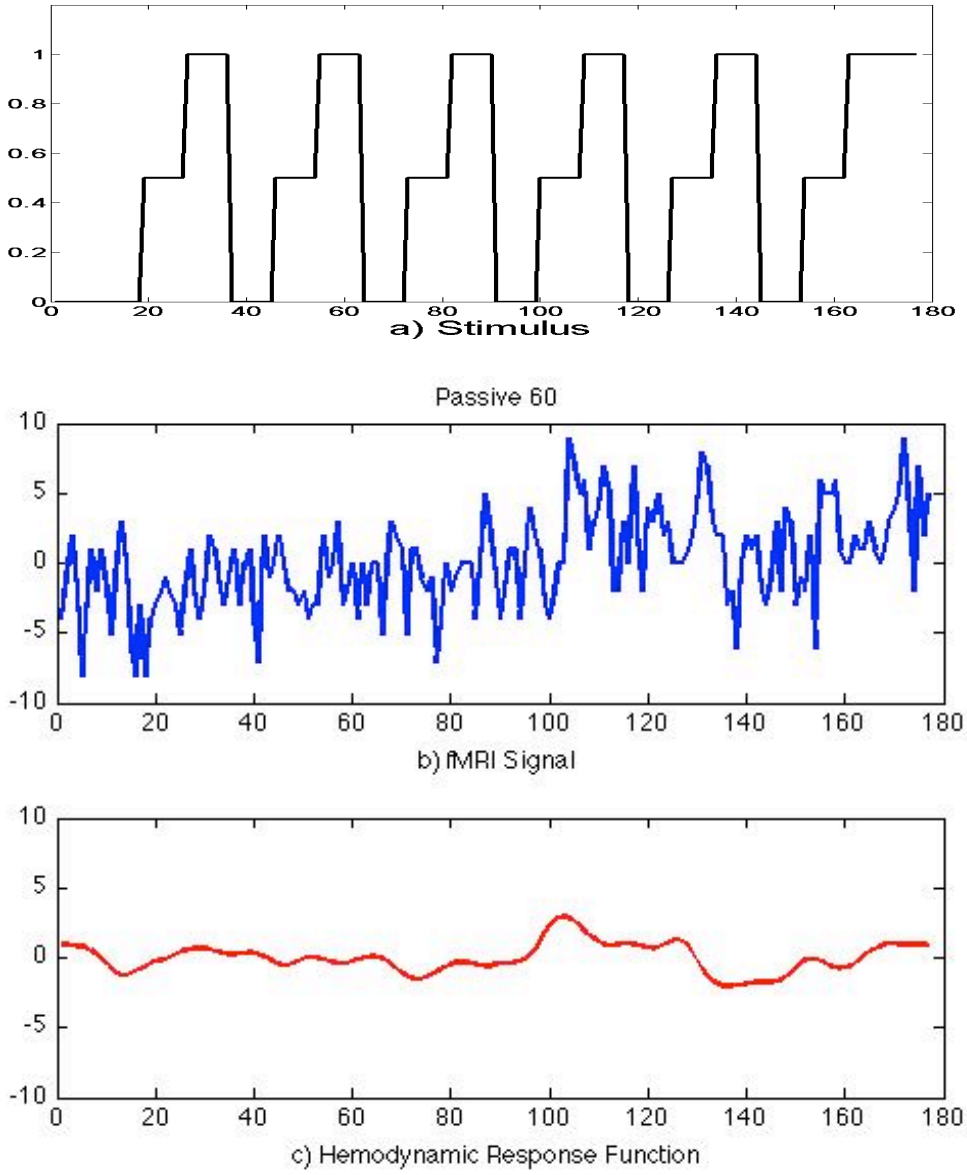


Figure 4.31 a) Applied stimuli b) fMRI Signal c) Hemodynamical Time Series obtained from 60th voxel of Passive_male_1

In Figure 4.31 there are also random variations in the hemodynamic time series. These random variations, as in Figure 4.30, are most probably due to the routine neural processes, which alters the fMRI intensity at every moment. Physiological factors may also cause these variations. Since the fMRI signal depends on the interaction between physiological factors (i.e. blood flow, blood volume, oxygen metabolism) any fluctuations and changes in these factors evoke neural activity even if they are not caused by the applied stimuli.

c) Motion Voxels:

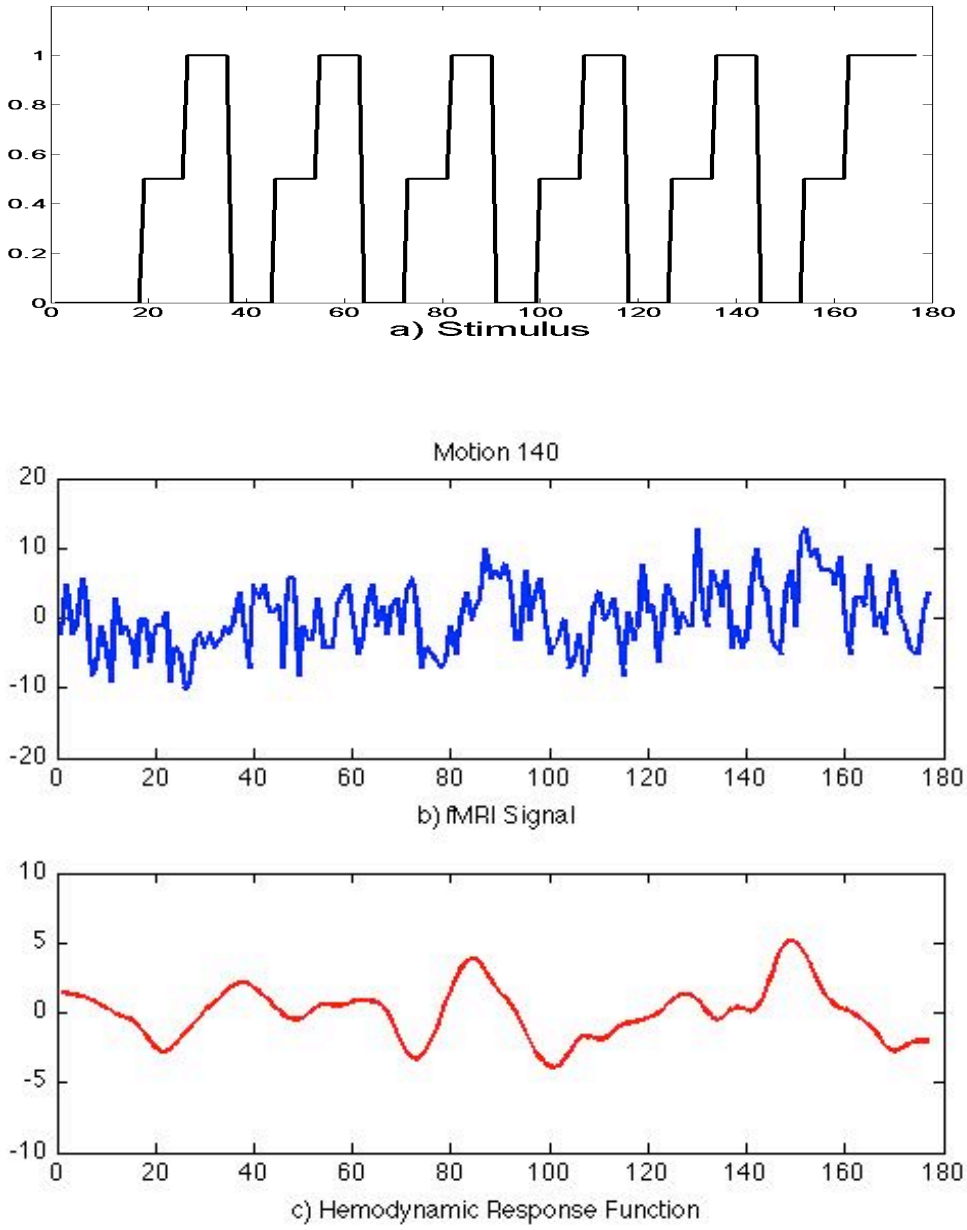


Figure 4.32 a) Applied stimuli b) fMRI Signal c) Hemodynamical Time Series obtained from 140th voxel of Motion_male_1

During an experiment, the subject may shift the position of his head; move his shoulders, arms, or legs to become more comfortable; and swallow more due to nervousness. These subject motions cause signal variability in fMRI studies. If the subject's head moves, then time series of each voxel is derived from more than one brain location. We named such voxels as "motion voxels". Because of the head motion, the hemodynamic responses of the neighbor voxels can be mixed. In the best cases, head motions are corrected during data preprocessing (See Appendix A). In the worst cases, it cannot be corrected and the hemodynamic time series may begin as active voxel but end up, after motion, as passive voxel or vice versa. In Figure 4.32 a motion voxel that is a mixture of active and passive voxels is observed. The activation peaks at about the third and the fifth stimuli representations indicate that there is activation at these durations. But in the rest of the experiment there is not a clear peak evoked due to the applied stimuli.

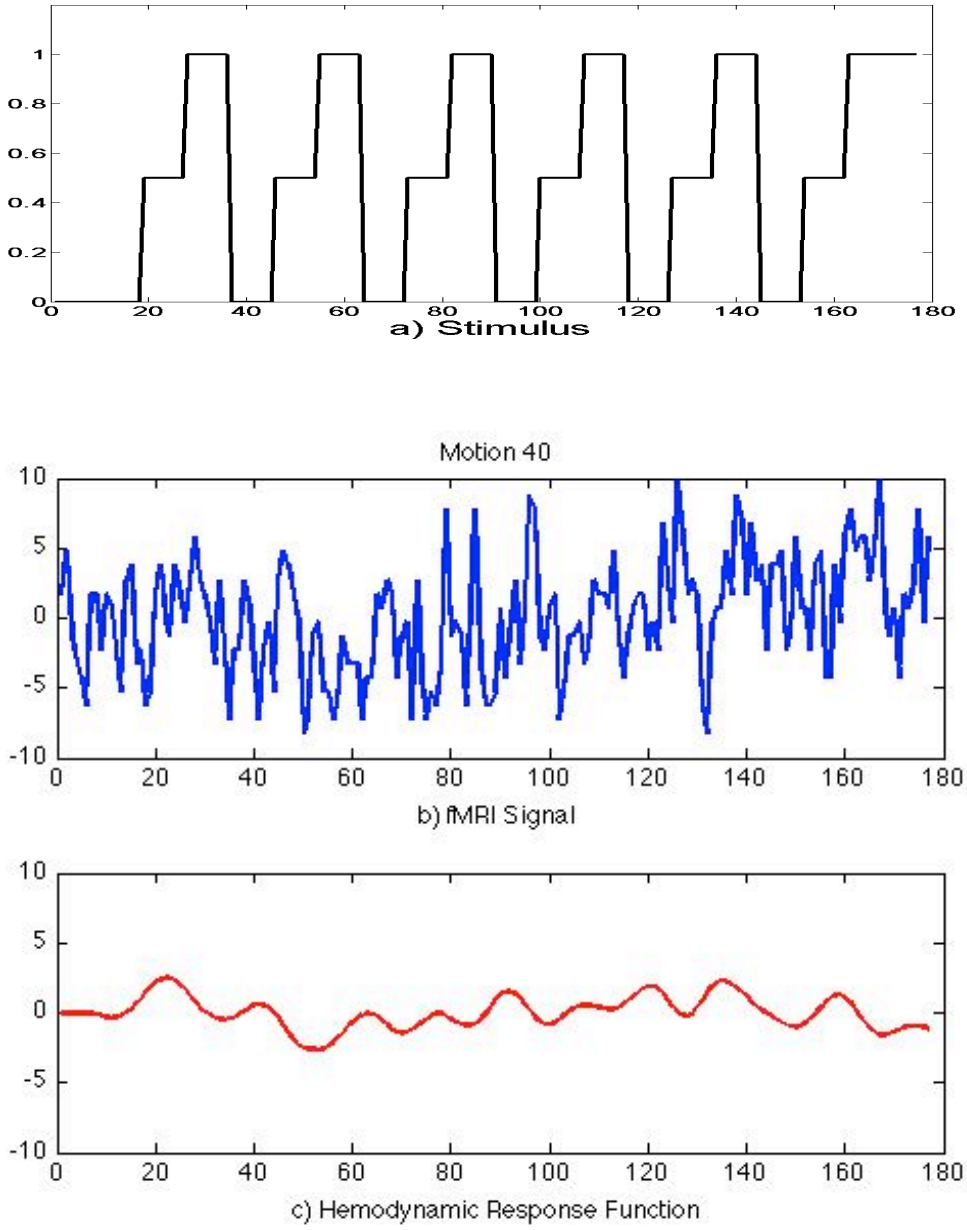


Figure 4.33 a) Applied stimuli b) fMRI Signal c) Hemodynamical Time Series obtained from 40th voxel of Motion_male_1

Figure 4.33 illustrates another type of motion voxel in which two passive voxels are mixed. At about the time point of 70, the baseline level of the hemodynamical time series slightly increases, but still exhibits noisy peaks. This points out that there are at least two different types of responses in this fMRI signal.

Besides head motion, the motion due to cardiac activity also causes variations in the fMRI signal. In particular, cardiac effects are mostly too fast to be sampled effectively. So the variability due to the cardiac activity is distributed throughout the fMRI time series in a manner that may be difficult to identify or correct. In Figure 4.33 the signal variations may also be caused due to the cardiac activity.

4.2 Clustering Results

After extracting the hemodynamical time series by means of MAP Blind Deconvolution, we group them according to their waveforms depending upon voxel activation. Since the shapes of the extracted hemodynamical time series due to the neural activation are different for active and passive voxels as we discussed in the previous section, we can cluster them using this underlying information. The modified Hausdorff distance which was investigated in details in Chapter 3, can successfully detect the similarities between these hemodynamical time series.

In order to demonstrate the steps our algorithm and test the performance of the method we propose, firstly, we propose to test a set of simulated BOLD signals based on the Balloon Model [80], [81], [82]. And secondly, we conduct experiments on a real fMRI data set.

4.2.1 Clustering Results for Simulated Data

The set of data in our simulations consists of simulated BOLD signals belonging to active voxels as well as the ones belonging to inactive voxels. The parameters for the simulation of BOLD signal change is the same as in [82], ε=0.5, τ_S= 0.8, τ_f= 0.4, τ_0= 1, α= 0.2, E_0 = 0.8, V_0 = 0.02. We use two different stimulus patterns to generate active and inactive voxels. These patterns are shown in the Figure 4.34.

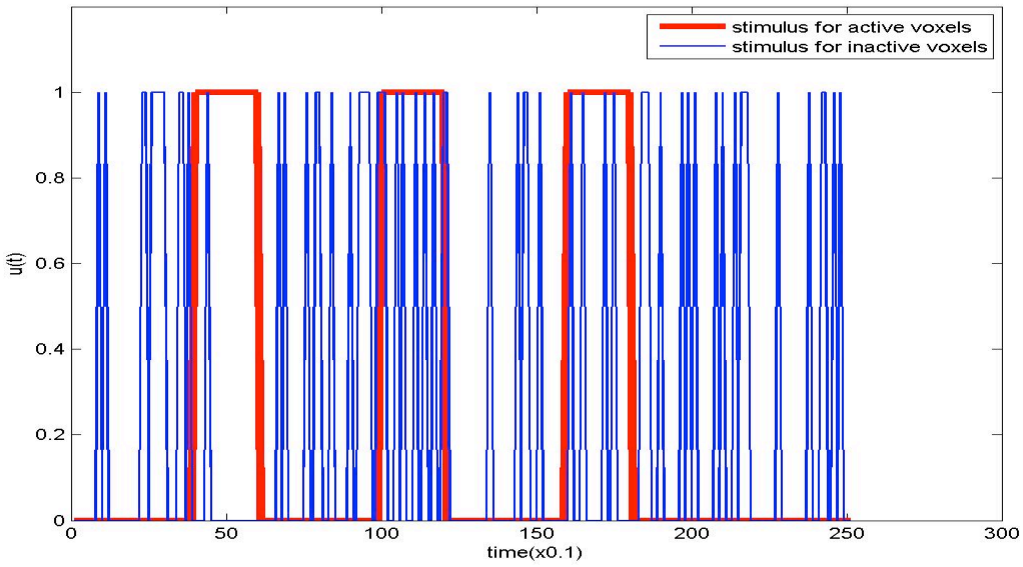


Figure 4.34 Stimulus patterns for active and inactive voxel simulations

For a simulation of a neural task we use a stimulus pattern which has the impulses in the intervals [40, 60], [100, 120] and [160, 180]. As for the simulation of BOLD signals of inactive voxels, ideally, we should have a zero stimulus. However, in order to have a better representation of inactive voxels due to the given specific task, we use a noisy stimulus to simulate the non-task related activities within the brain. An example of random stimulus for a passive voxel is illustrated in Figure 4.34. At every time interval for each passive voxel, we independently create a stimulus with probability 0.2 so that these random impulses simulate the activities within the brain that are not related with the experimental task.

An example of simulated BOLD signals for active and passive voxels is shown in the Figure 4.35:

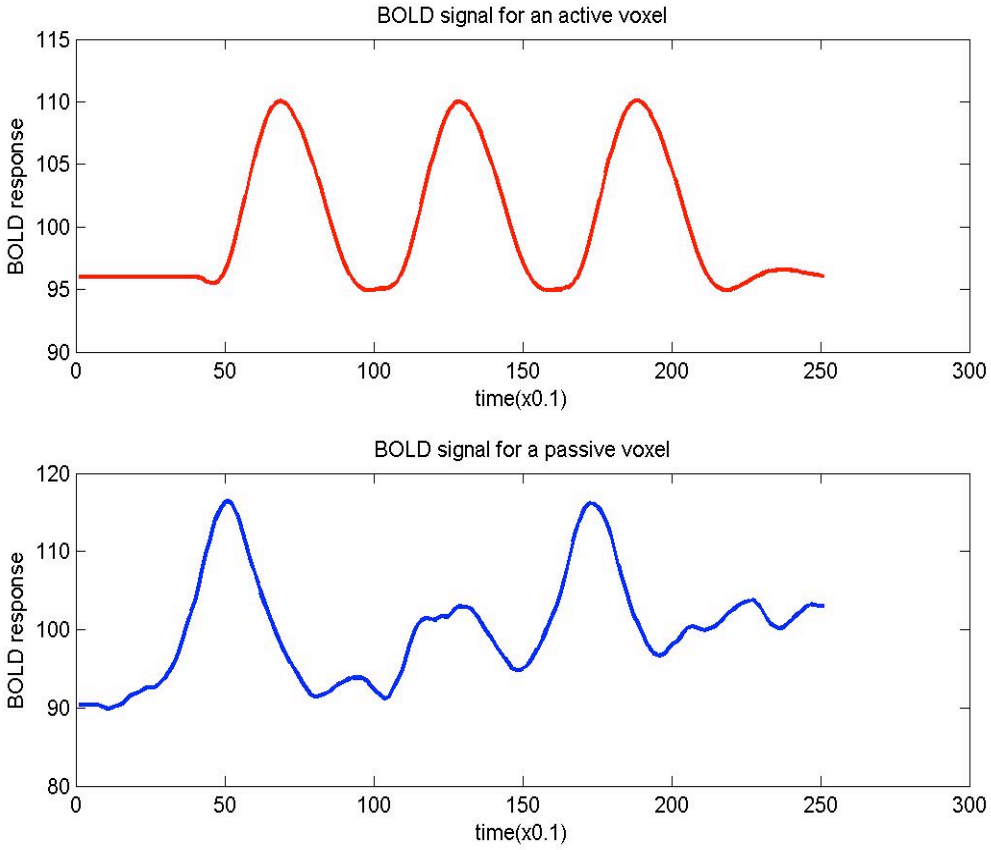


Figure 4.35 Examples of simulated active and passive BOLD signals

Generally, fMRI signals are output of several noisy processes due either to some intrinsic processes occurring in the brain or to the measurement process itself. For example the noise added onto the zero stimuli for modeling the BOLD response of passive voxels is an intrinsic noise. In order to have more realistic simulations and modeling of fMRI signals, we additionally use four more noise sources which are related to the measurement process.

(1) fMRI signals might have lag in time. We create a uniformly distributed lag for the BOLD signals, i.e,

$$x(t) \xrightarrow{\Delta \text{ lag in time}} x(t-\Delta) \text{ where } \Delta \sim Uniform([0,\sigma])$$

(2) fMRI signals might have drift in time. We create a quadratic random drift and add it onto the BOLD signal, i.e,

$$x(t) \xrightarrow{\text{quadratic drift in time}} x(t) + at^2 + bt \text{ where } a,b \sim Normal(0,\frac{\sigma^2}{N^2})$$

(3) fMRI signals might have noise which is mostly due to the noisy measurement. We create a Additive White Gaussian Noise (AWGN) to simulate such an effect, i.e,

$$x(t) \xrightarrow{AWGN\ noise} x(t)+n(t) \text{ where n(t)} \sim Normal(0,\sigma^2)$$

(4) Finally, fMRI signals might also have sampling jitter which is, like AWGN, mostly due to the jittery measurement process.

$$x(t) \xrightarrow{\text{sampling jitter}} x(t+\Delta(t)) \text{ where } \Delta\text{(t)} \sim Normal(0,\sigma^2)$$

In the demonstration of our algorithm, we use these parameters as:

$\sigma_{AWGN} = 4,\ \sigma_{jitter} = 4,\ \sigma_{LAG} = 16,\ \sigma_{DRIFT} = 16$

Figure 4.36 shows an example of the simulated a pair of active and passive fMRI signals:

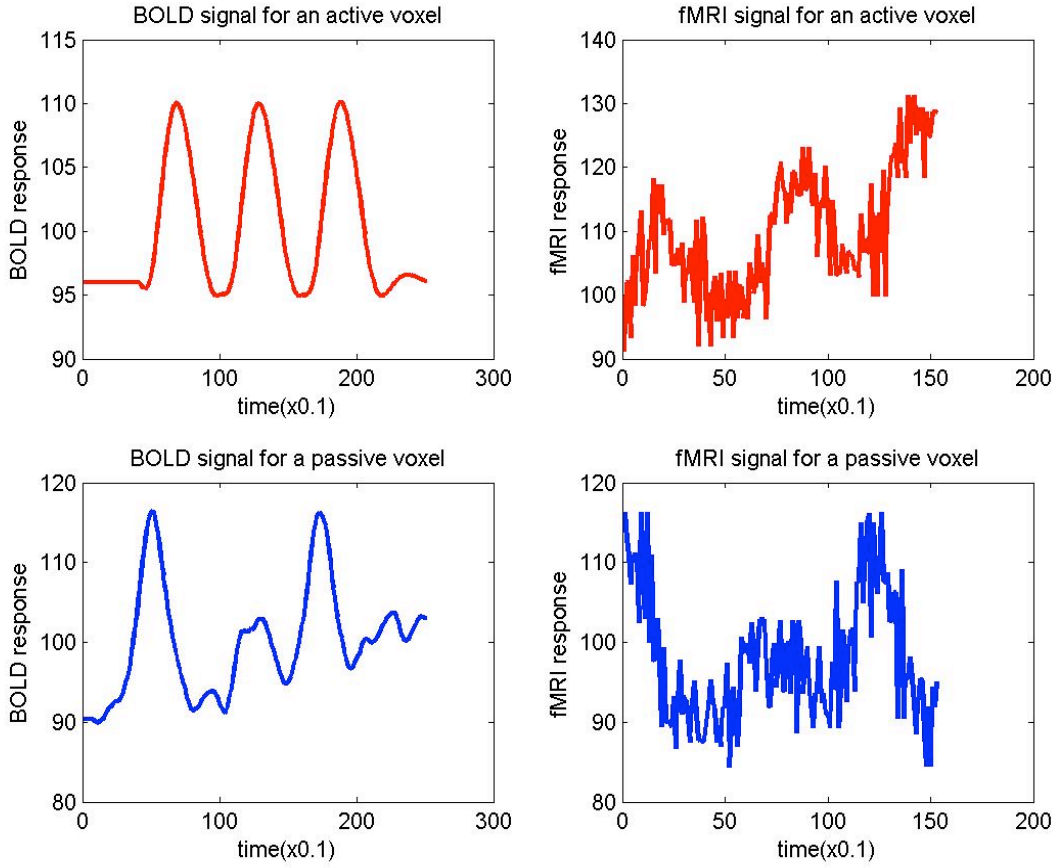


Figure 4.36 Examples of simulated active and passive fMRI signals

Based on this simulation model, we generate our simulated fMRI data set which consists of 1000 simulated fMRI time series data along with the true labels, i.e,.

$$R = \{r_i\}_{i=1}^{N} \subset \Re^{NxT}, Y = \{y_i\}_{i=1}^{N} \subset \{1,2\}^{Nx1}, N = 1000, T = 250$$

Each of the simulated fMRI is of dimensionality 250. If an fMRI is *'active'* or *'passive'* then it is labeled as 1 or 2 respectively. Locations of the active and passive fMRI's in our data set are as follows:

$y_i = 1$ if $i \in \{1,2,...,500\}$

$y_i = 2$ if $i \in \{501,502,...,1000\}$

In our algorithm we use the following parameter values:

- Blind deconvolution parameters:
  $\kappa = 0.1, p = 10$
- Modified Hausdorff Distance Parameters:
  $\alpha = 10$ (covering parameter)
  $\tau = 0.05$ (scaling of time)
- Spectral Clustering Parameters:
  $k = 6$ (number of neighborhood)
  nclass = determined by 'eigengap' heuristic

This particular setting of parameters is not necessarily the optimum one. Here, we use a reasonable set of parameter values and in the performance analysis in Chapter 5 we are modifying the parameters and analyzing the performance change due to these changes.

Recall that spectral clustering is a graph based method; it starts with constructing a graph on data and generating the matrix W which holds the edges and edge weights for each pair of graph nodes. There are several ways proposed for this construction such as fully connected graphs and partially connected graphs. For fully connected graphs the most common method is that each node is connected to every other node and the weight for each node is assigned according to a Gaussian kernel using Euclidean distance, $W(i,j) = \exp(-g * \left\| d_i - d_j \right\|^2)$. This is also called similarity matrix. For the partially connected graphs, the most common way is to adopt a neighborhood approach using the similarity matrix, i.e,

$$W(i,j)=\begin{cases}\exp\left(-g\left\|d_i-d_j\right\|^2\right) & \text{iff } d_i \text{ is in the k'th neighborhood of } d_j \\ 0 & \text{otherwise}\end{cases} \tag{4.4}$$

$$\begin{aligned}W(i,j)&=\left(\max W(i,j),W(j,i)\right)\\ W(i,j)&=W(j,i)\end{aligned} \tag{4.5}$$

Taking the maximum of two edge weights is to just make the similarity matrix symmetric. Using the minimum is also a possible approach. Instead of having a real valued similarity matrix, using a binary valued similarity matrix is also reasonable choice, i.e,

$$W(i,j)=\begin{cases}1 & \text{iff } d_i \text{ is in the k'th neighborhood of } d_j \\ 0 & \text{otherwise}\end{cases} \tag{4.6}$$

$$\begin{aligned}W(i,j)&=\left(\max W(i,j),W(j,i)\right)\\ W(i,j)&=W(j,i)\end{aligned} \tag{4.7}$$

In this work, we used a binary valued similarity matrix with respect to modified Hausdorff distance as in (4.4) and (4.5). That is to say, we detect whether a point d_i is in the k'th neighborhood of d_j by using the modified Hausdorff distance. Once having the similarity matrix constructed, then spectral clustering solves the generalized eigenvalue problem $Lv=\lambda Dv$ where **L** is the Laplacian and is **D** the diagonal matrix computed via similarity matrix **W**.

In Figure 4.37, we show the first 6 eigenvectors of this equation when all the eigenvectors are sorted in ascending order with respect to their eigenvalues.

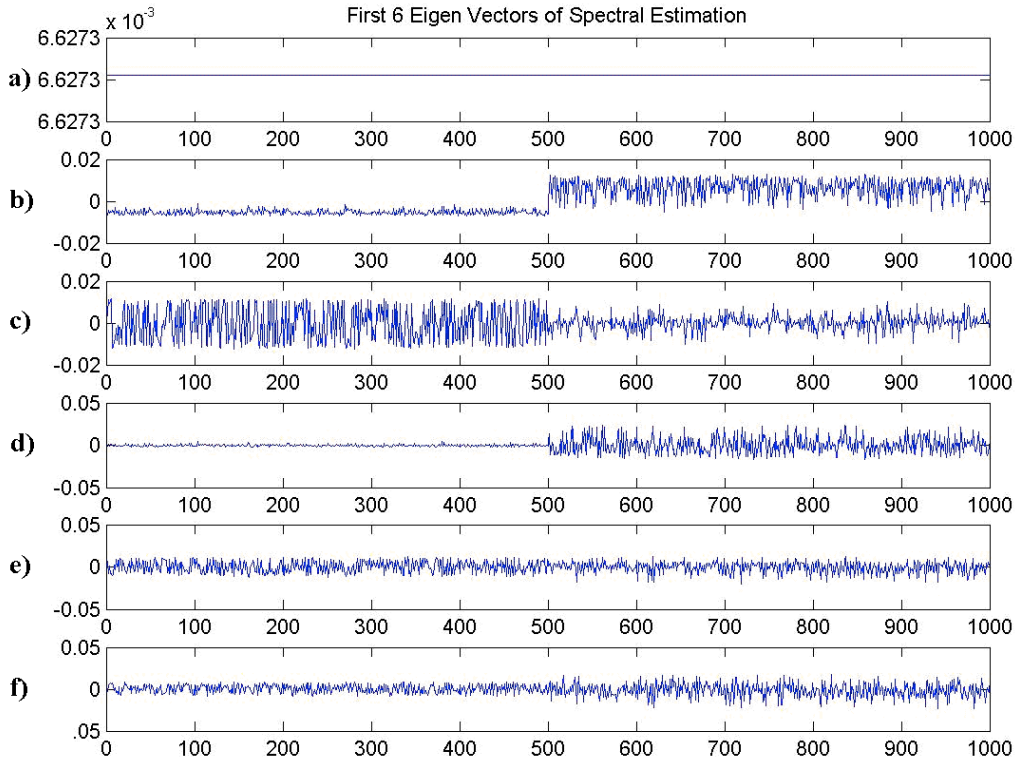


Figure 4.37 First 6 eigenvectors of Spectral Estimation

First eigenvector in Figure 4.37a is basically a constant one, so provides no discriminative capability for clustering and activation detection. However, on the second eigenvector, Figure 4.37b, the two classes become visible, based on the step that the eigenvector starts taking greater values after the instant 500. This is an expected result because by design we generated two classes in our data set and using the eigenvectors as many as the number of expected clusters in the data is what is done, in general, in spectral methods. Furthermore, on the 3rd and 4th eigenvectors, Figure 4.37c and Figure 4.37d, clusters get separated even better. Hence using first four eigenvectors together should give a very good clustering. After the 4th eigenvector, for instance using the 5th

and 6th eigenvectors, Figure 4.37e and Figure 4.37f, bring no gain in terms of clustering quality.

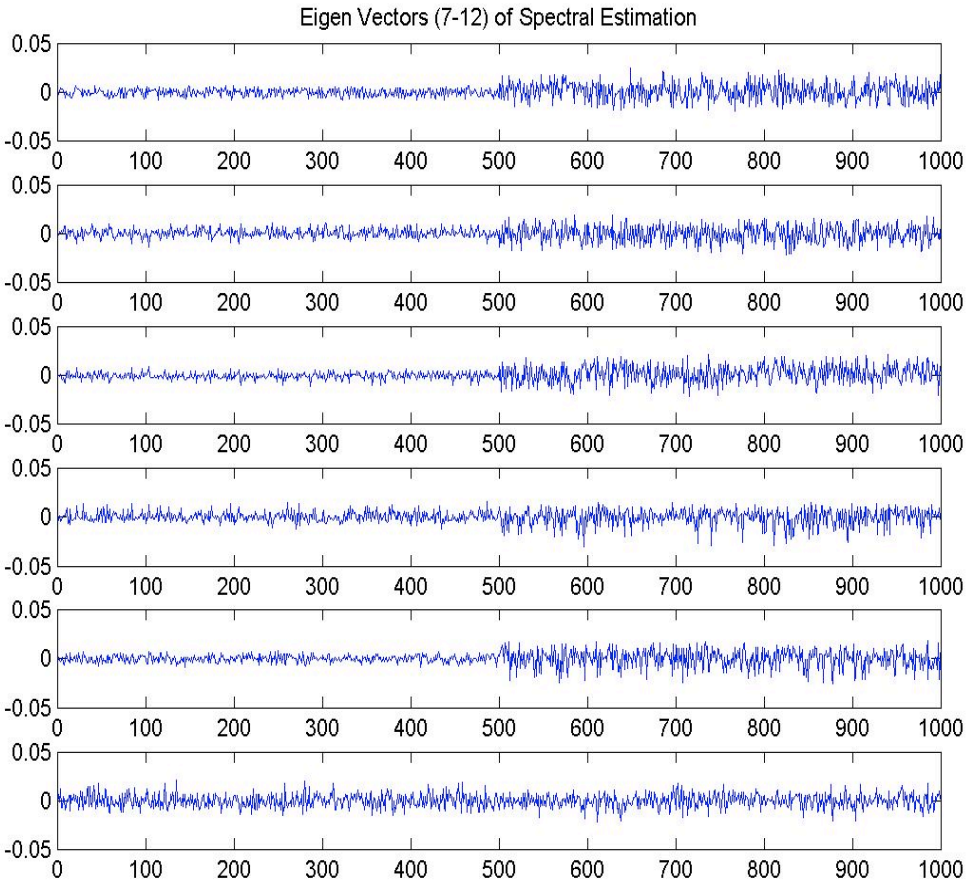


Figure 4.38 7.-12. Eigenvectors of Spectral Estimation

However, for the eigenvectors following the sixth one there is no separation and bringing them to the eigenmap would bring only confusion. Note that in Figure 4.38, the two classes are not visible exactly. Figure 4.39 shows the corresponding eigenvalues:

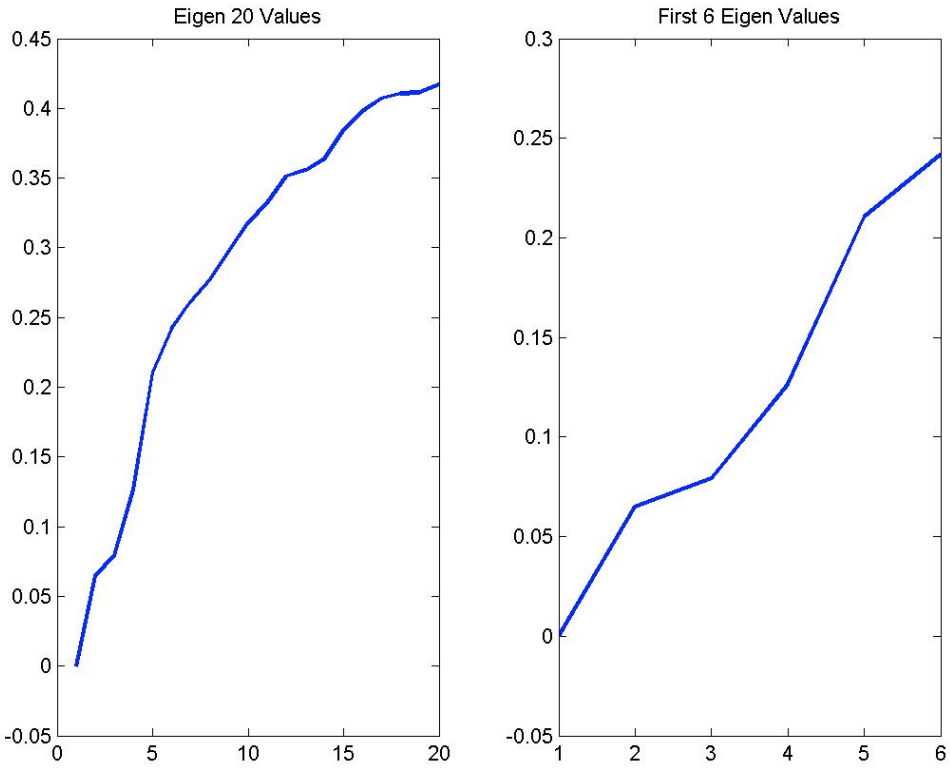


Figure 4.39 Eigenvalues of eigenvectors

Deciding upon the number of clusters is always an important issue in clustering. For spectral clustering method there is what is called in the literature 'eigengap' heuristic, which is a good way of estimating the number of clusters after the generalized eigenvalue problem is solved. Eigengap heuristic tells that when the data is perfectly clustered, then $\lambda_1, \lambda_2, ..., \lambda_{nclass} = 0$, $\lambda_i >> 0$, $i > nclass$. Since no real data is perfectly clustered, when there is a sudden increase, so-called 'eigengap', on the plot of eigenvalues, that should be the number of clusters. In Figure 4.39, and on right, we show the eigenvalues of the first 6 eigenvectors. It is clear that from 3rd to 4th, the eigenvalue jump is more than 50%. On the other hand, from the 2nd eigenvalue to 3rd eigenvalue there is a very little change. Hence, this justifies that we clearly have 2 or 3 clusters, nclass=2, 3. We can use both nclass = 2 and nclass = 3. Note that by design we had 2

classes in our data, however, we impose large deviations onto the original BOLD responses through sampling jitter, lag, drift, noise, which can further scatter the data into a few new classes. Hence we can also use as nclass = 3. The spectral clustering is doing a decent job together with the modified Hausdorff distance under these clustering numbers as shown below.

A final note on this, Figure 4.40 shows the spectral-mapped data distribution using the eigenvectors 2, 3 and 2, 3, 4.

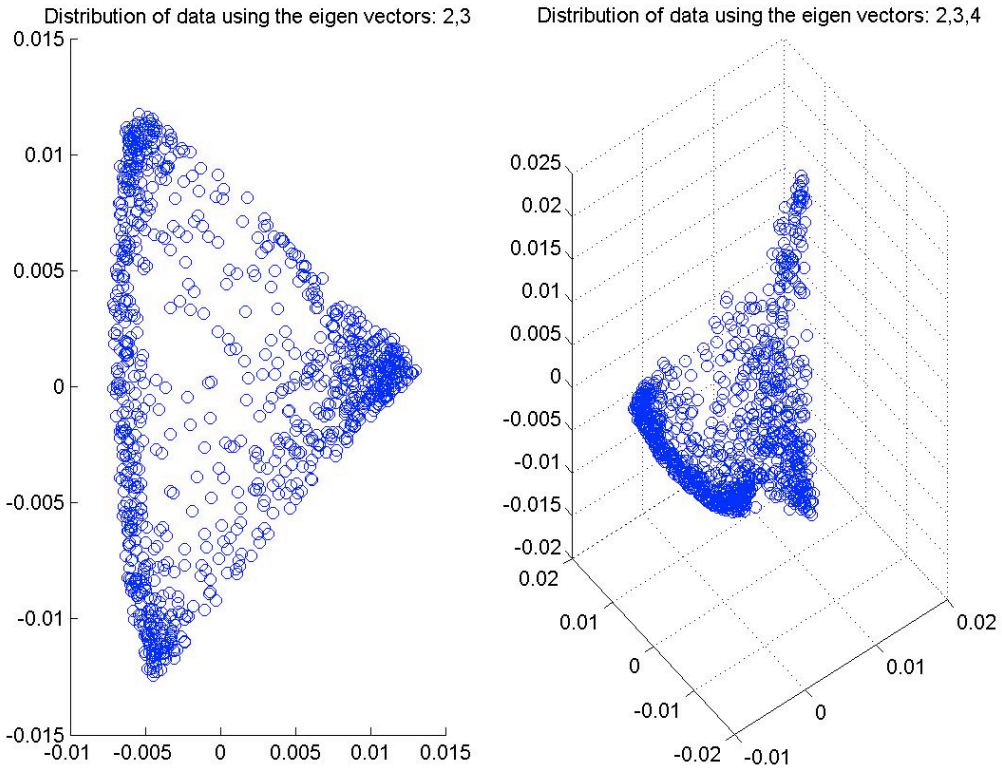


Figure 4.40 Different distributions of the simulated data

According to Figure 4.40, data are scattered onto a triangular area. Using the eigenvectors 2 and 3, a two class separation is perfectly visible. One class is on an edge

(the line: x=-0.005) of the triangular area and the other class is located on the corner (corner: (0.01, 0)) of that area. These are probably the classes of active and passive fMRIs which are the most visible and best separated. Also, on the edge, we clearly have two clusters on two ends of the edge; these are brought by the 3rd eigenvector. Even though we expect two clusters in the data scattering; we obtain an extra one when we use the 3rd eigenvector. This is due to the several noise conditions we applied on our simulation model. However, separation between the classes active and passive is much more powerful than between the ones under imposed noise and model conditions (noise, lag, drift, and jitter). As a result, at the earliest stage (using the 2nd eigen vector) the active and passive classes immediately become visible. Afterwards, the other clusters in the data come into scene.

Once the spectral mapping is finalized as $(x_i \leftarrow \mathbf{EV}(i,2:nclass))$, we locate the clusters by Expectation Maximization (EM) clustering algorithm. In
Figure 4.41 we show the clustering results illustrated on 2D plots:

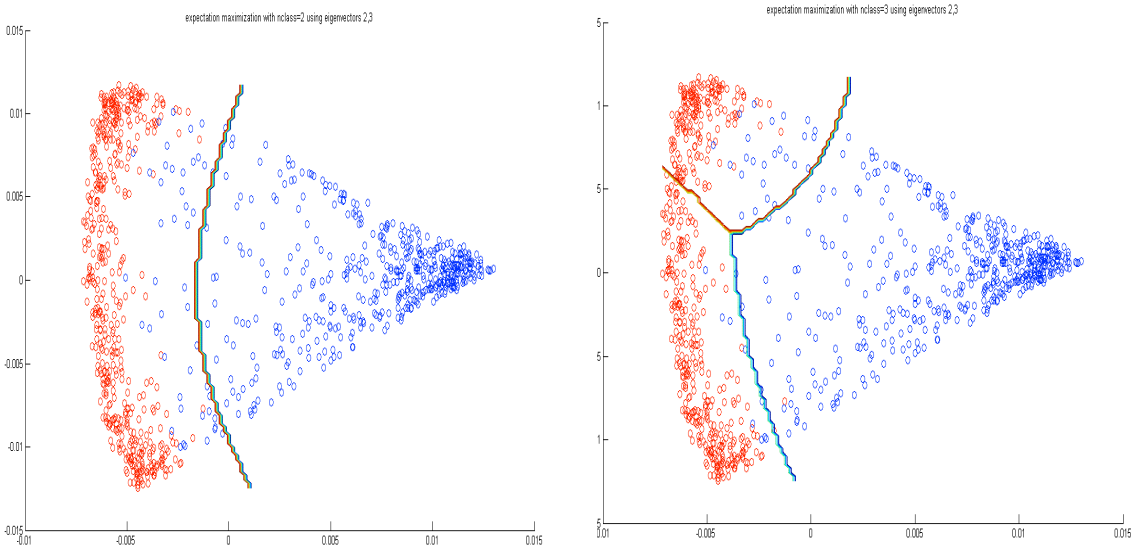


Figure 4.41 Different clustering results on 2D plots

By means of spectral clustering, we can perform active and passive detection in both cases (nclass = 2 and nclass = 3). However, we detect two modes existing in the active class if we set nclass = 3.

If we use nclass = 2, then the sensitivity (percentage of active fMRI's that are correctly detected) for active-passive detection turns out to be: **1**. And the specificity (percentage of passive fMRI's that are correctly detected) is **0.9040**. For nclass = 3, sensitivity: **0.9980**; sensitivity: **0.9320**.

Furthermore, one should note that above clustering is performed under an unbalanced operating point. That is to say, prior probabilities for active and passive classes are not necessarily 0.5. On the other hand, we actually do have equal prior probabilities, i.e, 500 data points from both classes. One can always change this (when EM clustering is used) by pushing the decision boundary returned by EM clustering towards one of the classes whichever is more desired such that the decisions will not be biased to one of the classes or can be biased in a desired way. For instance, by setting p(active) = 0.45 and p(passive)=0.55, one can possibly increase the specificity and decrease the sensitivity. A plot of sensitivity and specificity values versus all possible choices for operating points is named 'Receiver Operating Characteristics' (ROC) curve. By incorporating with this curve, one can adjust the balance between sensitivity and specificity regarding the requirements of his/her application. For instance, cancer / HIV detection problems require an extremely high specificity in order to avoid the devastating effect of telling a patient that he has the disease when actually he has not. In such cases, by choosing the right operating point, one can have high specificity while sacrificing from the sensitivity. Following is the ROC curve for our activation detection problem:

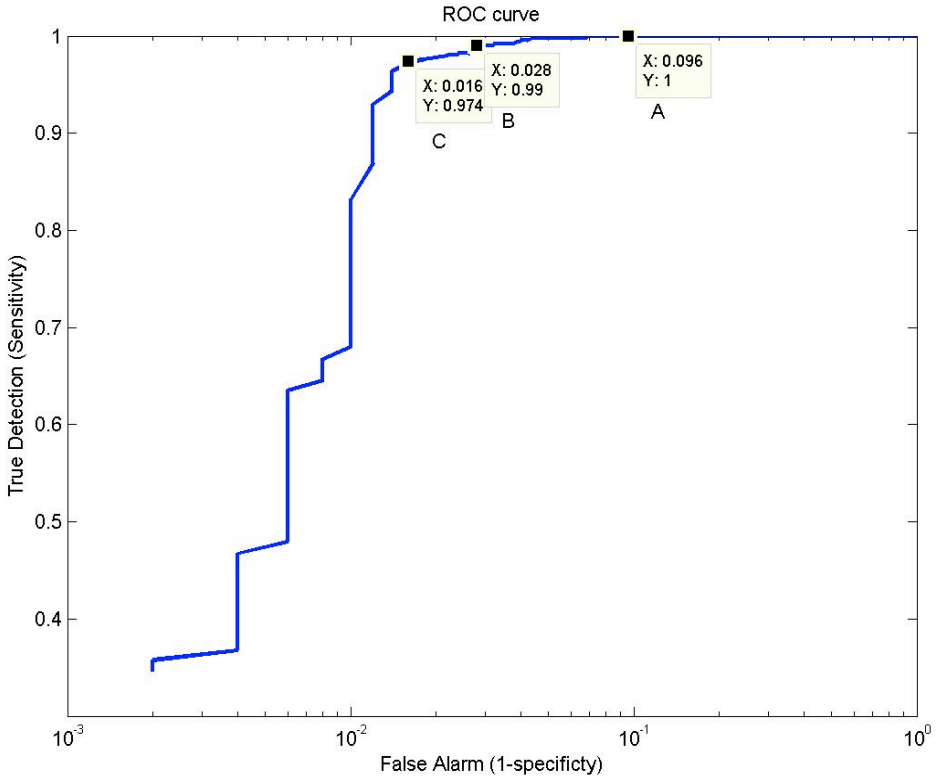


Figure 4.42 ROC Curve for simulation data with $\sigma_{AWGN} = 4$, $\sigma_{jitter} = 4$, $\sigma_{lag} = 16$, $\sigma_{drift} = 16$

The original sensitivity / specificity values we reported previously correspond to the operating point A in the ROC curve as illustrated in Figure 4.42 for our simulation data. Note that Specificity is 1 – False Alarm. For the operating point A, sensitivity is 1 and specificity is 0.9040. If we choose to operate at point B, sensitivity drops to **0.99** (only a **1%** drop) however, on the other hand we gain **6.8%** increase in specificity and it becomes **0.972.** And similarly for the operating point C, for a 2.6 % drop in sensitivity, we gain an 8% increase in specificity. Hence, for our activation detection problem, actually operating point B is better than the operating point A. However, in order to have a fair comparison among all cases (as opposed to finding the best operating point for each of the cases), we do not touch the boundaries returned by EM clustering and show the results without optimizing them in this chapter. To keep in mind that, all the results can always be improved by finding the right operating point with the help of the ROC

curves. However for an fMRI application, the number of voxels in a region of interest which becomes activated in response to a task condition is not arbitrary, so p(active) and p(passive) can not be optimized. Usually we operate with a rate of 1-2% of the voxels being active.

As shown, even though we use relatively large deviations from the original BOLD responses, we still do have high sensitivity and specificity. At the heart of this is the hemodynamic response function estimation through blind deconvolution and the modified Hausdorff metric. Figure 4.43 shows the clustering of the same simulated data when the hemodynamic response function is not estimated and raw fMRI is directly used with nclass = 3 and 4 eigenvectors.

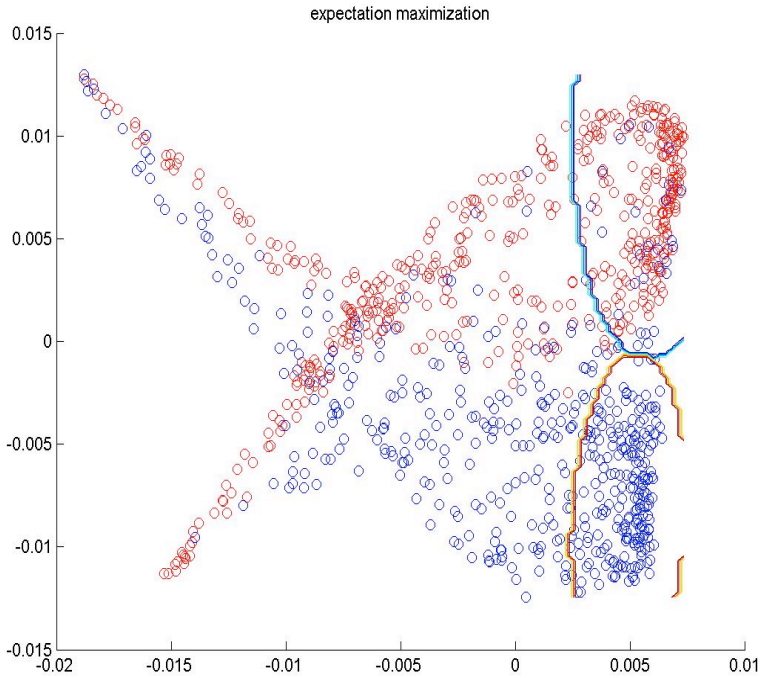


Figure 4.43 Clustering on the raw fMRI data (without HRF extraction)

In this case, sensitivity is 0.7960 and the specificity is 0.5800. This proves the necessity of HRF estimation and shows that our blind deconvolution for HRF estimation performs very well. Table 4.1 summarizes our findings so far:

Simulation results	**Sensitivity**	**Specificity**
Spectral clustering of HRF's through modified Hausdorff distance	**1**	**0.9040**
Spectral clustering of raw fMRIs through modified Hausdorff distance	0.7960	0.5800

Table 4.1 Clustering results using and BOLD response and the raw fMRI

Having shown the efficiency of our HRF estimation through MAP blind deconvolution, we also compare our clustering results with the results of the cases when other distance functions are used such as Euclidean distance, Mahalanobis distance, cosine similarity, RBF similarity, Cross Correlation based distance. In Table 4.2, we show the sensitivity and specificity values corresponding to the cases different metrics used in Spectral Clustering. In all cases we used the optimum number of clusters according to the eigengap heuristic.

Simulation results	Modified Hausdorff distance	Euclidean distance	Mahalanobis distance	Cosine Similarity	RBF similarity	CC based distance
Sensitivity	**1**	1	0.0300	1	1	0.9800
Specificity	**0.9040**	0.7540	0.9820	0.8460	0.7540	0.8360

Table 4.2 Comparison of different distance and similarity measures

Among these distances, our **modified Hausdorff distance** turns out to be **the best one** with the sensitivity and specificity values in Table 4.2. Among the other distances, Euclidean distance and RBF similarity do not perform well mainly because they do not consider any possible lags in the data. Since Cosine Similarity and CC based metric perform similarly and performs better than Euclidean distance, we can conclude that correlation based features for fMRI data analysis provides better active/passive clustering than directly using Euclidean metric. In particular, Mahalanobis distance fails on active/passive clustering. This is because Mahalanobis distance is good only for the cases when the data is normally distributed, i.e, when the data is scattered around a center. However, our data has two clusters and so two centers: Active Cluster and Passive Cluster. In other words, our data has a distribution similar to a mixture of Gaussians with two components. For this reason, Mahalanobis distance cannot detect the clusters in our data.

Next we analyze how our method behaves under different conditions. For that, we design simulation sceneries such that the effect of the noise conditions are considered separately: we add individually one at time, AWGN noise, sampling jitter, lags, and drift. Table 4.3 shows the results on these different conditions, and in each case we use either nclass = 2 or nclass = 3 whichever gives a better clustering.

		AWGN	Sampling Jitter	Lag	Quadratic drift
Sigma=1	Sensitivity	1	1	1	1
	Specificity	1	0.9920	0.9880	0.9200
Sigma=2	Sensitivity	1	1	1	1
	Specificity	0.9800	0.9920	0.9860	0.9420
Sigma=4	Sensitivity	0.9980	1	1	1
	Specificity	0.9680	0.9860	0.9640	0.9200
Sigma=8	Sensitivity	0.9980	1	0.9400	1
	Specificity	0.9380	0.97	0.9820	0.9160
Sigma=16	Sensitivity	0.9560	0.9100	1	1
	Specificity	0.8460	0.8000	0.888	0.8880

Table 4.3 Clustering results under different noise conditions

According to our findings summarized in Table 4.3, our method is very robust to AWGN and sampling jitter. It performs very well even under the large noise conditions $\sigma_{AWGN} = 8,$ and $\sigma_{jitter} = 8$.

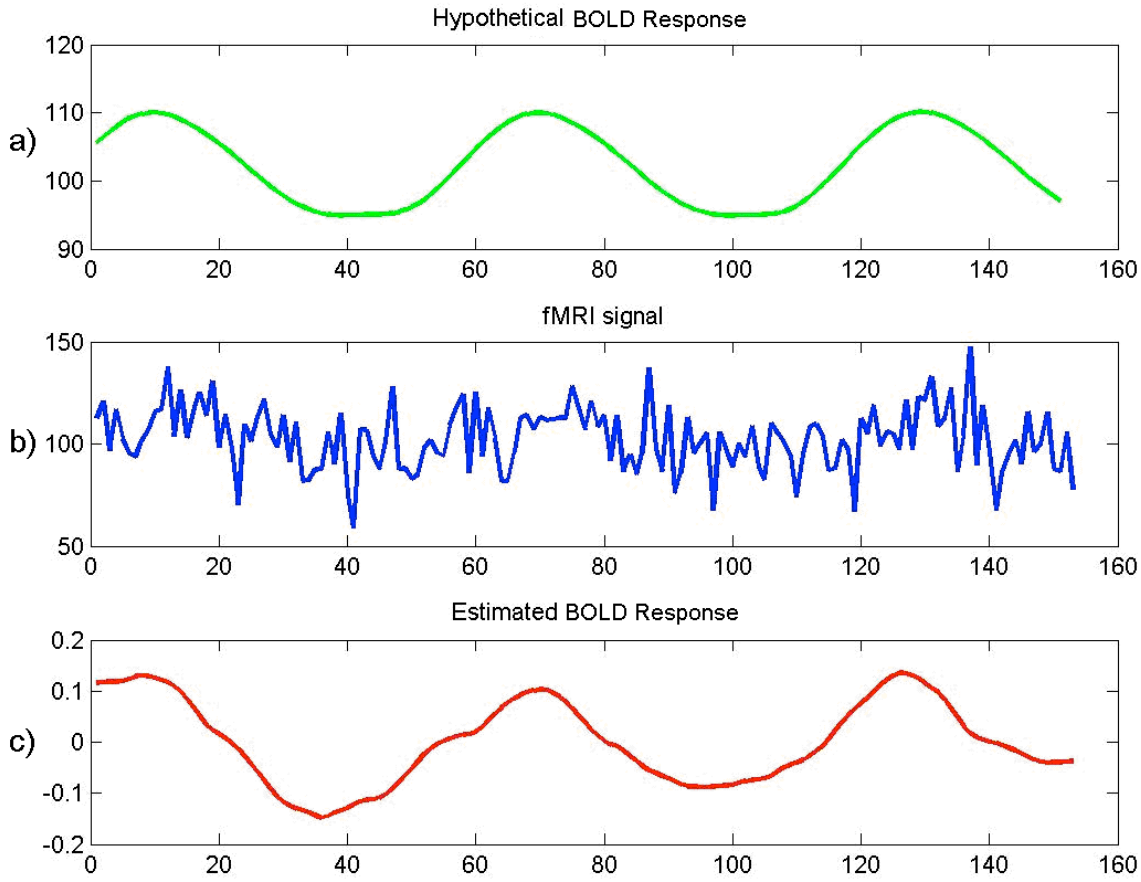


Figure 4.44 An example fMRI with AWGN noise of standard deviation 16, the hypothetical BOLD Response, and the estimated BOLD Response

Figure 4.44b shows an example fMRI with AWGN noise of standard deviation 16, the hypothetical BOLD Response, Figure 4.44a, and the estimated BOLD Response, Figure 4.44c. MAP blind deconvolution is seen to successfully estimate the BOLD Response. Although we put a large amount of AWGN noise, it is visually clear that our estimated BOLD Response is close to the hypothetical one.

The situation is a little different for the cases when we have lag and quadratic drift. In the case of quadratic drift, the distances in the passive cluster, the pair-wise distances do not change significantly because there is already a large variety in that class and drifts do not bring extra deviations since we estimate and filter quadratic drifts in the phase of

preprocessing of fMRI. Consequently, the data scattering within the passive class do not change. However, this is not the case for the active class. Note that we have generic BOLD response for the active class and we add quadratic drift onto that generic BOLD response. Then in our preprocessing step, we filter out the drifts. However, even very small drift estimation errors in the preprocessing dramatically change the pair-wise distances because basically before the drifts those pair-wise distance were all 0 (d(BOLD response, BOLD response) = 0). As a result, those small drift estimation errors become very important when compared to the original pair-wise distances, which are 0. Also noting that quadratic drifts take effect on the fMRI signals globally, the only source of the variety in the active class becomes only the drift estimation error. For this reason, the whole class of active fMRIs can be parameterized over only the drift amount. Figure 4.45 illustrates the clustering under $\sigma_{drift} = 16$.

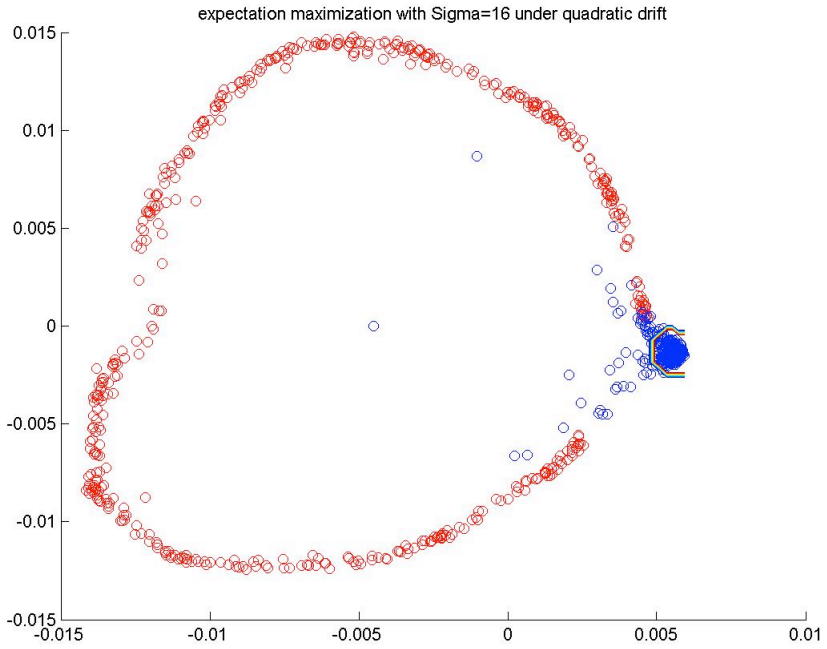


Figure 4.45 Clustering results with $\sigma_{drift} = 16$

As a result of the aforementioned reasons, the active class get scattered on almost a circular path. And since the passive class does not have a similar powerful structure, it gets cluttered around a center. Noting that the active class has a lot larger variance than the one of passive class, EM clustering draws the boundary tightly around the center of passive class and so does not allow the ~10% of the passive class come into the detected passive cluster. However, if we shift the boundary a little towards the active class, we would improve the specificity a lot without sacrificing from the sensitivity since there is a very good separation between two classes even though EM draws the boundary conservatively. We can argue similarly for the other cases of quadratic drift. As the drift strength decreases, the variance in the active class decreases as well, and so the variance between of two classes gets more balanced and so the specificity gets better. In particular, this case of simulations provides a very good example of where one can incorporates with the ROC curve and improve the clustering performance by choosing the right operating point. Figure 4.46 gives the ROC curve:

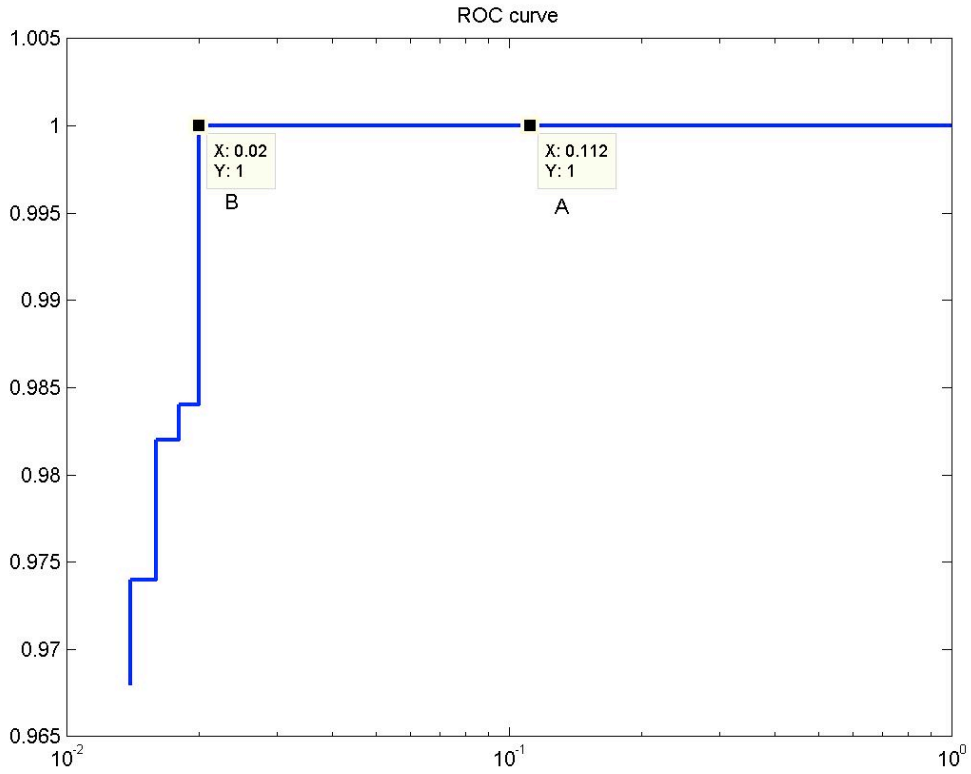


Figure 4.46 ROC Curve for simulation data with $\sigma_{drift} = 16$

According to the ROC curve in Figure 4.46, the results for this case presented in Table 4.3 correspond to the clustering operating at point A which is, indeed, not so preferable choice. By choosing the operating point B instead, we can nicely push the boundary between both classes in favor of the passive class and so improve the specificity almost **10%** without sacrificing from the sensitivity at all. This turns into **100% sensitivity and 98% specificity**. Although this is the case where we used the largest amount of drift, we still have near-perfect results. This proves that our drift estimation and filtering in the phase of preprocessing works very well and so our method is greatly robust to quadratic drift.

Now let us consider the case of adding lags, if $\sigma_{lag} = 16$ then we add random lags with varying strengths up to 16. Then in this case, we basically have 16 different clusters in

our data since we do not have continuous lags but discrete ones. So each fMRI with some lag gets precisely coupled with the ones having the same amount of lag. The eigenvalue distribution shows this very clearly in Figure 4.47:

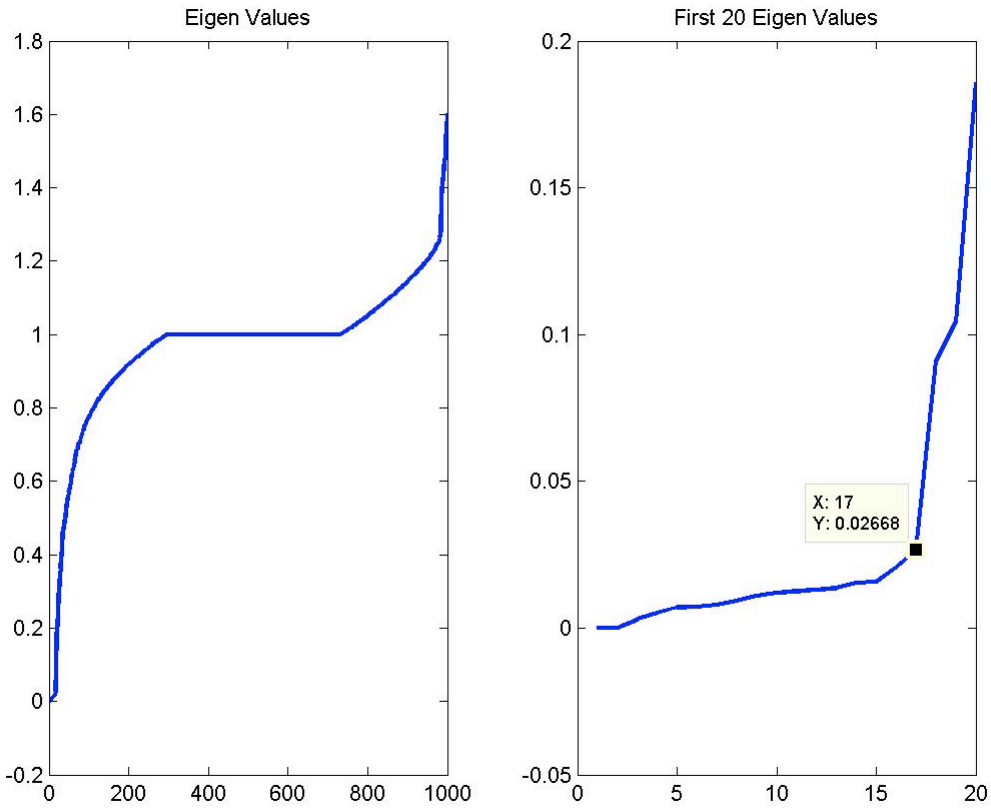


Figure 4.47 Eigenvalues when $\sigma_{lag} = 16,$

As it is shown in Figure 4.47, there is large eigengap between the 17^{th} and 18^{th} eigenvalues. This justifies our thought that there should be 17 clusters. Following is the clustering of the data under this condition; note that we have two-dimensional plot, so not all clusters are visible:

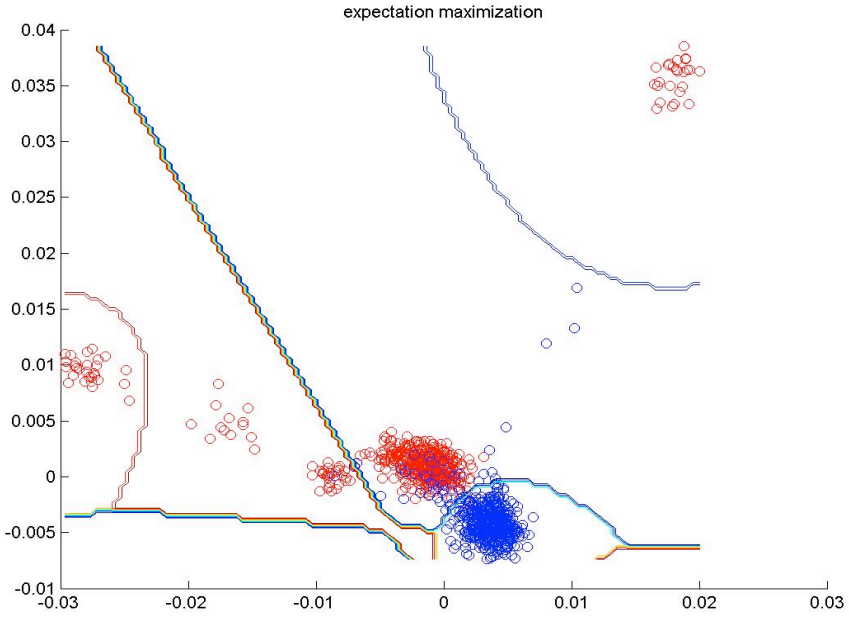


Figure 4.48 Clustering of data into 17 clusters

As it is clear from the Figure 4.48 that, as the strength of the lag is increased, the data is get clustered into more clusters and our method successfully locate those clusters.

Next, we present our sensitivity and specificity results under combined noise and lag-drift conditions in Table 4.4.

	Sensitivity	Specificity
Sigma_AWGN = 2; Sigma_Jitter = 2 Sigma_Drift = 2; Sigma_Lag = 8	1	0.9440
Sigma_AWGN = 4; Sigma_Jitter = 2 Sigma_Drift = 2; Sigma_Lag = 8	0.9980	0.9300
Sigma_AWGN = 8; Sigma_Jitter = 2 Sigma_Drift = 2; Sigma_Lag = 8	0.9200	0.9420
Sigma_AWGN = 4; Sigma_Jitter = 4 Sigma_Drift = 2; Sigma_Lag = 8	0.9600	0.9620
Sigma_AWGN = 4; Sigma_Jitter = 8 Sigma_Drift = 2; Sigma_Lag = 8	0.9120	0.9540
Sigma_AWGN = 4; Sigma_Jitter = 4 Sigma_Drift = 4; Sigma_Lag = 8	0.9980	0.9360
Sigma_AWGN = 4; Sigma_Jitter = 4 Sigma_Drift = 8; Sigma_Lag = 8	0.9700	0.9580
Sigma_AWGN = 4; Sigma_Jitter = 4 Sigma_Drift = 16; Sigma_Lag = 8	0.9960	0.9120
Sigma_AWGN = 4; Sigma_Jitter = 4 Sigma_Drift = 16; Sigma_Lag = 16	1	0.9040
Sigma_AWGN = 8; Sigma_Jitter = 4 Sigma_Drift = 16; Sigma_Lag = 16	0.9780	0.8240
Sigma_AWGN = 8; Sigma_Jitter = 8 Sigma_Drift = 16; Sigma_Lag = 16	0.7900	0.8620
Sigma_AWGN = 16; Sigma_Jitter = 8 Sigma_Drift = 16; Sigma_Lag = 16	0.5800	0.7740
Sigma_AWGN = 16; Sigma_Jitter = 16 Sigma_Drift = 16; Sigma_Lag = 16	0.6080	0.6200

Table 4.4 Clustering results under combined noise and lag-drift conditions

Based on our findings presented in Table 4.4, we can conclude that up to the noise and jitter with $\sigma_{jitter} = 8, \sigma_{AWGN} = 8$ our method stays robust and performs reasonably well. On the other hand, even if we use large amount of delays and drifts, since we incorporate delays in our distance calculation and drifts in our data preprocessing, our method stays above 90% sensitivity and specificity. As already discussed before, these results can be improved a lot if the ROC curve is further incorporated with and the right operating point is chosen. Here, we used a balanced clustering between both classes, that is to say, we assumed prior probabilities for both classes are 0.5 in order to observe the effect of our noise conditions and have a better comparison among them.

4.2.2 Clustering Results for Real Data

We also conduct experiments on real fMRI data. The set of data that we use in our experiments consists of in total of 510 fMRI time series along with the true labels, i.e, $V = \Re^{NxT}, Y = \{1,2,3\}^{Nx1}; N = 510, T = 177$. Different from our simulations, we have three classes in this part: *'active', 'passive', 'motion'* among which the *'motion'* class is newly added. This actually consists of fMRI's coming from some passive voxels, however, because of the some small-scale motion artifacts on fMRI's such as regular oscillatory activity of the heart and lungs, this set of fMRIs do not look like the other 'passive' fMRI's. We still consider them as 'passive' though in our experiments since the voxels that they are registered for are 'passive'. Hence, it is basically another independent cluster in the passive class and leading to running our algorithms with at least nclass = 3, but when calculating the sensitivity and specificity we label them as 'passive'.

Each of the time series is of dimensionality 177 and if an fMRI is labeled as 'active', 'passive' or 'motion' then the corresponding labels are 1, 2, and 3 respectively. Locations of these labeled fMRI's in our training data are as follows:

$y_i = 1, i \in \{1,2,...,180\};$

$y_i = 2, i \in \{181,182,...,330\};$

$y_i = 3, i \in \{331,332,...,510\}$

For this data set, we use the following parameter values:

- Blind deconvolution parameters:
  $\kappa = 1, p = 10$
- Modified Hausdorff Distance Parameters:
  $\alpha = 10$ (covering parameter)
  $\tau = 0.05$ (scaling of time)
- Spectral Clustering Parameters:
  $k = 6$ (number of neighborhood)
  nclass = determined by 'eigengap' heuristic

This particular setting of parameters is not necessarily the optimum one. Here, we use a reasonable set of parameter values and in the performance analysis chapter we will vary values to assess system performance to those changes. The only difference in this parameter setting when compared to the previous one is that κ=1 in this experiment. This is because the underlying neural task of this experiment involved more and complicated (2-folded) impulse periods. So, by decreasing the parameter as κ=1 we get a less smooth input signals.

Figure 4.49 is the eigenvector distribution for this data returned by our algorithm:

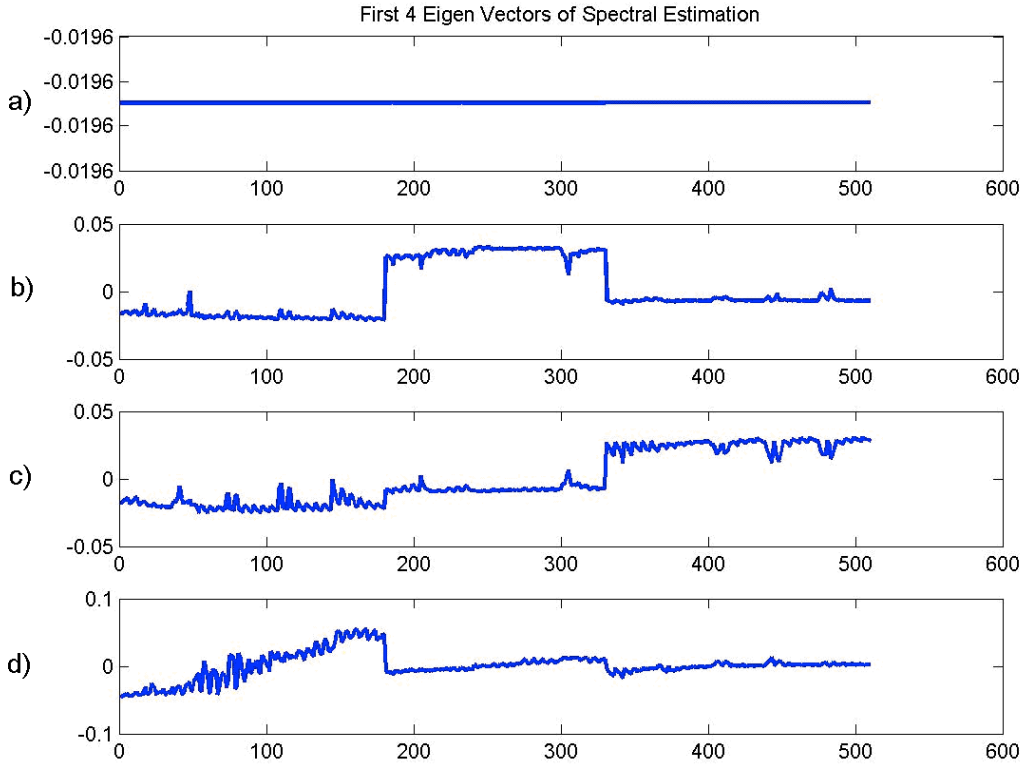


Figure 4.49 First 4 eigenvectors of spectral estimation of real data

First eigenvector in Figure 4.49a, like in our simulations, is a constant one, so it gives no information about the clusters. However, on the second eigenvector, Figure 4.49b**,** the 'passive' class of fMRI's becomes almost perfectly visible. And similarly, on the third eigenvector, Figure 4.49c**,** we finally get the 'motion' class of fMRI's. On the other hand, the 4th eigenvector, Figure 4.49d, is confusing the classes so we should not use it. Figure 4.50 shows the corresponding eigenvalues:

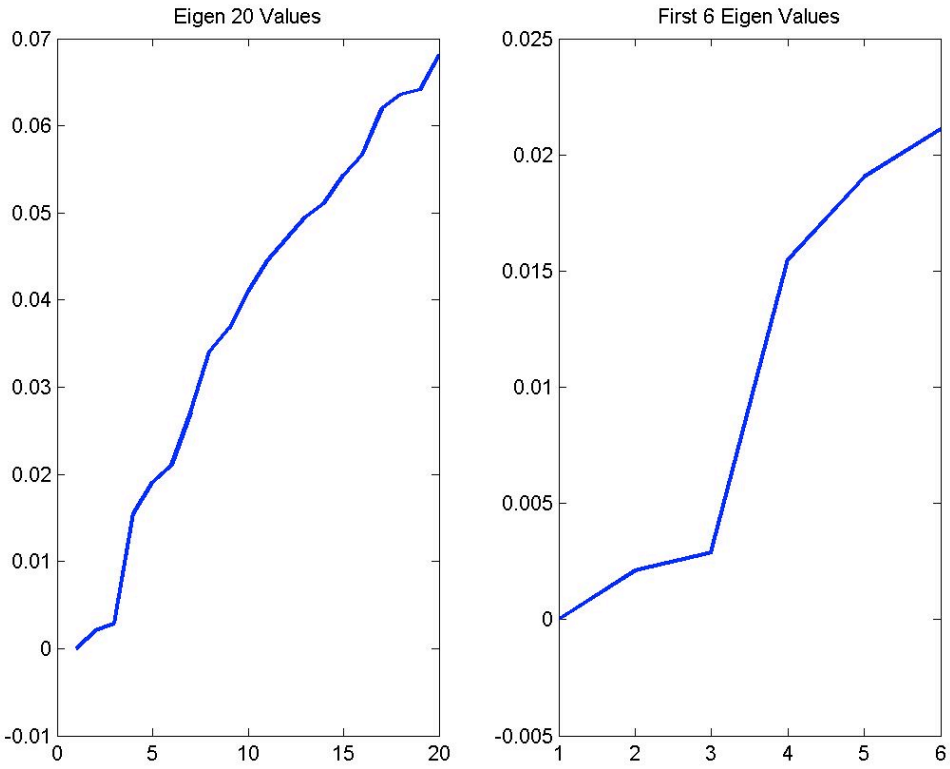


Figure 4.50 First 20 and 5 Eigenvalues of the eigenvectors respectively

According to the eigengap heuristic, the number of clusters in this data set should be 3 since there is a jump from the 3rd eigenvalue to the 4th one. This is what we expect because we know that there are three classes of fMRI's in our data: 'active', 'passive' and 'motion'.

Figure 4.51 shows the spectral-mapped data distribution using the eigen vectors 2, 3 and 2, 3, 4.

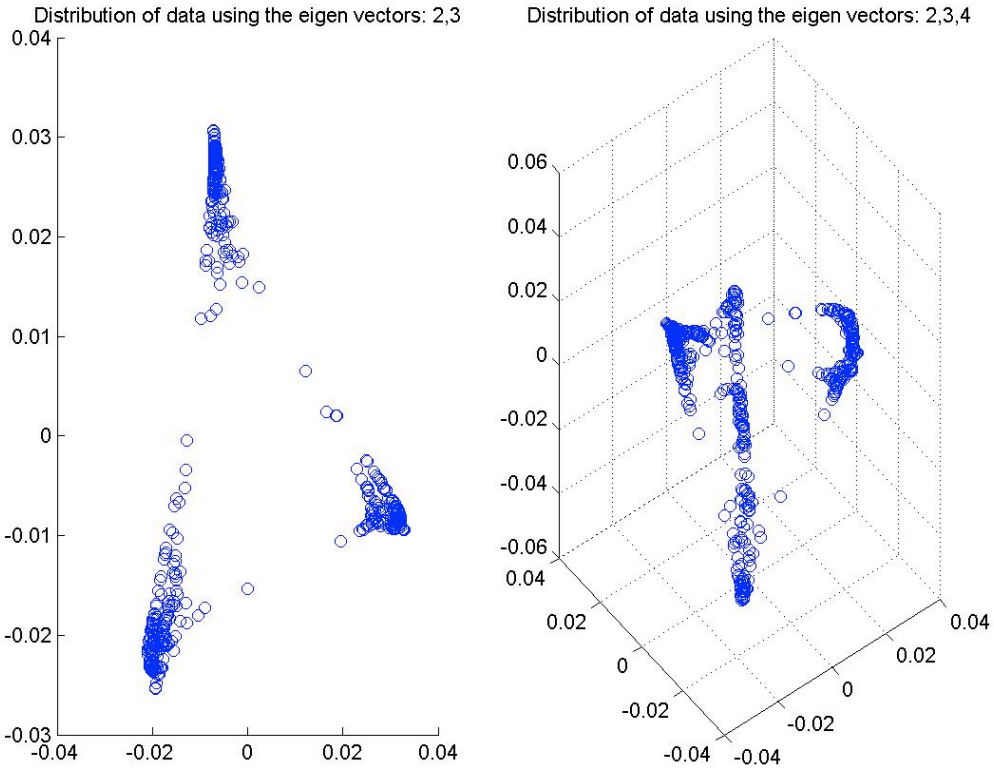


Figure 4.51 Distribution of the real data using different eigenvectors

As it is clear from the figure that using the 2nd and 3rd eigenvector, we recover all clusters very clearly, but on the other hand, if we further use the 4th eigenvector, the clusters become less separated. Hence, using the 2nd and 3rd eigenvector along with nclass = 3 should give a good performance. Then we run our EM clustering algorithm on this spectrally distributed data, which is illustrated in Figure 4.52:

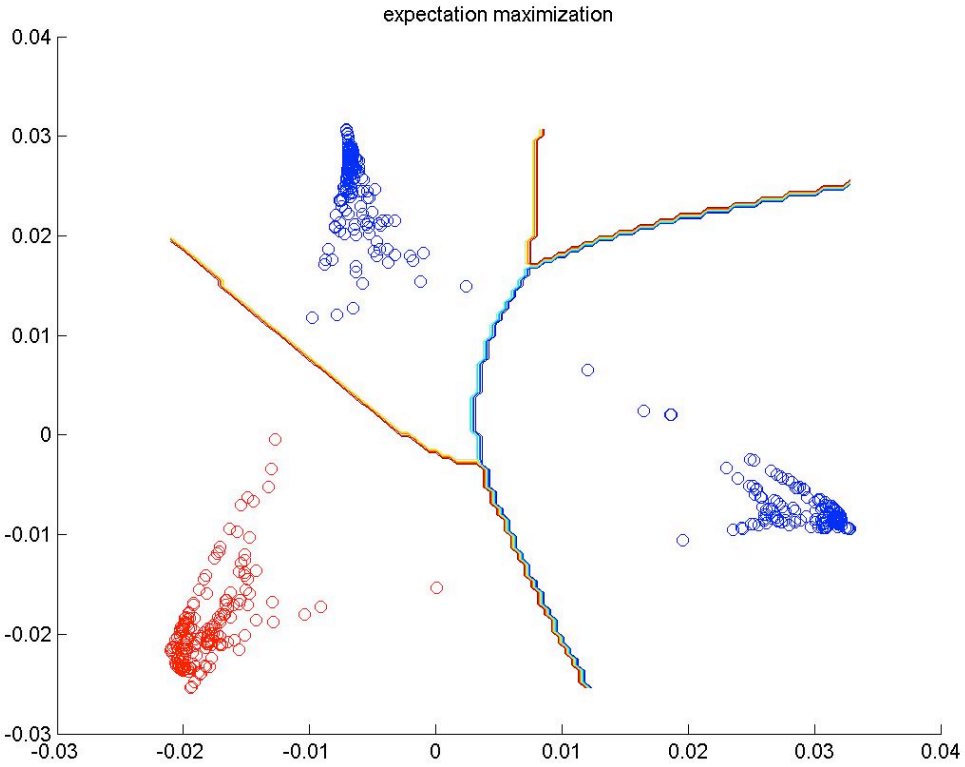


Figure 4.52 EM Clustering of the real data

As it is shown in Figure 4.52, our algorithm gets all three classes very nicely. Active class of fMRI's are well separated from the passive class of fMRI's which further includes the motion class of fMRI's as another cluster. **The sensitivity and the specificity turn out to be 1 in this case.** This shows, under real life conditions, our method not only successfully detects activation among brain voxels but also implicitly concentrate on the good features regarding the activation providing a very good separation between active / passive classes.

CHAPTER 5

5 SENSITIVITY AND PERFORMANCE ANALYSIS

In this chapter we analyze the sensitivity of our algorithm based on our simulation as well as the real data and through this analysis we optimize the parameters of our algorithm. Recall that our algorithm incorporates with the following parameters:

- Blind deconvolution parameters:
  κ (smoothess parameter)
  p (length of the convolution filter)
- Modified Hausdorff Distance Parameters:
  α (covering parameter)
  τ (scaling of time)
- Spectral Clustering Parameters:
  k (number of neighborhood)
  nclass (number of clusters determined by 'eigengap' heuristic

In the following we present our findings on the simulation as well as the real data sets. For each of these data sets, we start with a certain set of parameter values and then by changing each of them within an interval, we calculate the sensitivity and specificity.

Sensitivity Analysis on Simulation Data

Similar to Chapter 4, we generate a set of simulation data including all noise conditions but now with parameters:

$$\sigma_{AWGN} = 8, \sigma_{jitter} = 8, \sigma_{LAG} = 16, \sigma_{DRIFT} = 16$$

This setting is different from the one in the previous chapter. Here, on purpose, we use AWGN and jitter having standard deviations two times greater since we want to have a harder data set for sensitivity analysis to see the effect of the parameters to a better degree.

And we initially set the parameters of our algorithm as:

$\kappa = 0.01$
$\alpha = 10$
$\tau = 0.05$
$k = 16$

Table 5.1 shows the performance of our algorithm under different choices for the length of the convolution filter, *p*.

	Sensitivity	**Specificity**
p = 1	0.9960	0.9160
p = 5	0.9980	0.8920
p = 10	1	0.8800
p = 15	1	0.8600
p = 20	1	0.8540
p = 30	1	0.8580
p = 40	0.9980	0.8760
P = 60	0.8440	0.5340

Table 5.1 Sensitivity and specificity of the algorithm under different choices for *p*

According to our findings, as *p* increases the sensitivity does not change much, so that an increase from 0.9960 to 1 is not significant. On the other hand, specificity decreases. And when p is set to 60, then the algorithm turns out to be performing poorly. This is certainly expected. Considering the active/passive clustering, what is distinguishing between a passive fMRI and active fMRI is, the actual stimulation. For active fMRI's there is a common activation regarding the stimulus, as for the passive fMRI's, this is not the case. Ideally for passive fMRI's there should be no activation at all, however, due to resting state activities in brain, there still undergo a stimulated effect that can be modeled as randomly, which exists in our simulations as well. Recall that we randomly generate an impulse at every time instant with probability 0.2 for passive fMRI's. Hence the source of separation between the active fMRI's and passive fMRI's is the pattern of the underlying stimulus. First recall that as *p* increases we switch from estimating a hemodynamical time series to HRF estimation in which we do not have information about the stimulus pattern. In other words, as we increase *p*, we start having a better

estimate for the stimulus pattern as the convolution filter. If we use the HRF as the input to our clustering, the clustering will have no access to the stimulus pattern. And the performance decreases. The sensitivity does not decrease as fast as the specificity because the single-peak hemodynamics for active fMRIs are same or very similar to each other, so they are mostly clustered well even when p=60. On the other hand, since the variety within the class of passive fMRI's is large, they start mixing with active fMRI's quicker and the specificity decreases faster. Figure 5.1 is an illustration of our discussion:

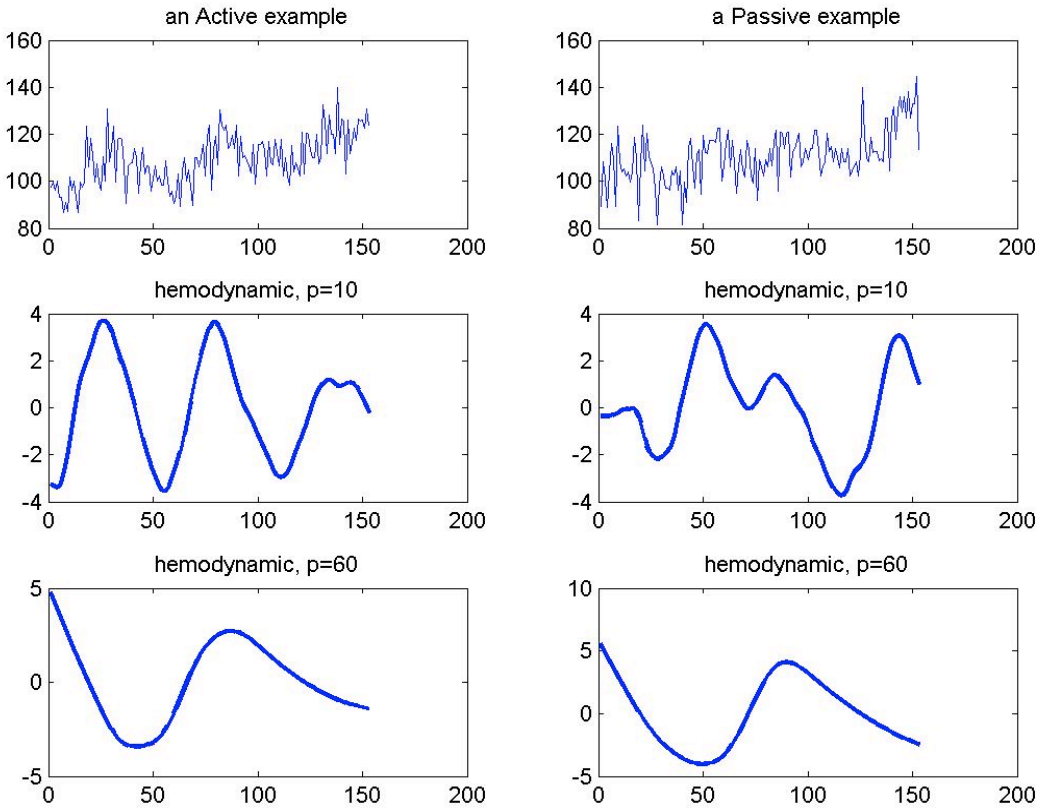


Figure 5.1 The effect of the parameter p on the estimations of active and passive voxel responses

In Figure 5.1 note that, when p = 10, we clearly see the duration and strengths of the stimulus on hemodynamics. On the other hand, when p = 60, the effect of stimulus on

the hemodynamic is separated and reflected on the other component on the blind deconvolution. As a result, using the HRFs in our clustering, as p increases, we loose the separation between active and passive classes.

Next, we show the effect of the parameter κ on our algorithm together with the following parameter values:

$p = 10$
$\alpha = 10$
$\tau = 0.05$
$k = 16$

Note that we choose as p=10, since it give a 100% sensitivity together with a good specificity. Table 5.2 shows the performance of our algorithm under different choices:

	Sensitivity	**Specificity**
$\kappa = 0.01$	1	0.8800
$\kappa = 0.05$	1	0.8480
$\kappa = 0.1$	1	0.8640
$\kappa = 0.5$	0.9960	0. 9260
$\kappa = 1$	0.9440	0.9260
$\kappa = 2$	0.9600	0.8980
$\kappa = 5$	0.99	0.8820
$\kappa = 10$	0.9840	0.8260

Table 5.2 Sensitivity and specificity under different κ values

The parameter κ of our blind deconvolution algorithm is to impose smoothness on estimated hemodynamics. The smaller it is, the smoother estimate we get for the hemodynamics. The optimum value turns out to be 0.5 for values greater than which the algorithm performs poorly.

Figure 5.2 shows the hemodynamics for $\kappa = 0.5$ and $\kappa = 5$.

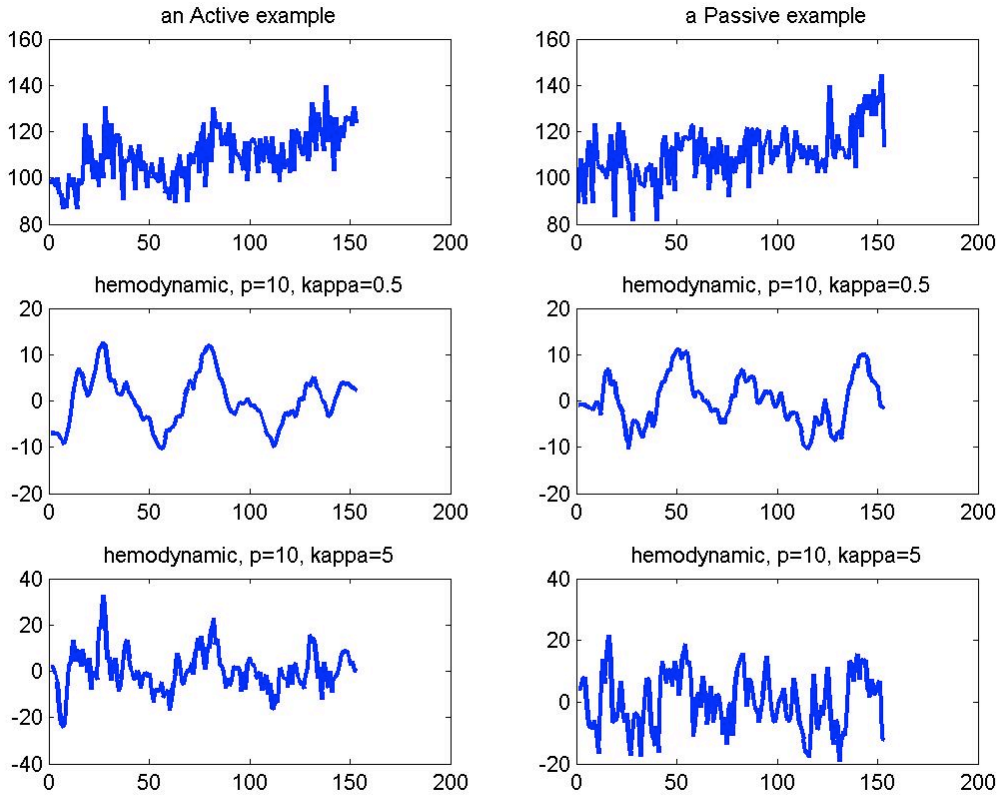


Figure 5.2 The effect of the parameter κ on the estimations of active and passive voxel responses

As seen when κ is too large, like 5, then the estimated hemodynamics become too noisy which badly affects our clustering. Our findings show that using κ as 0.5 is reasonable.

Next, we show the effect of the parameter α on our algorithm together with the following parameter values:

$\kappa = 0.1$
$p = 10$
$\tau = 0.05$
$k = 16$

Note that we choose as κ = 0.1, since it give a 100% sensitivity together with a good specificity. Table 5.3 shows the performance of our algorithm under different choices:

	Sensitivity	**Specificity**
$\alpha = 1$	0.9420	0.8360
$\alpha = 5$	0.9240	0.8660
$\alpha = 10$	0.9260	0.8560
$\alpha = 15$	0.9340	0.8320
$\alpha = 20$	0.9480	0.8180
$\alpha = 30$	0.9120	0.8440
$\alpha = 40$	0.9380	0.8140

Table 5.3 Sensitivity and specificity under different α values

According to our findings, our algorithm is not really sensitive to this parameter **on our simulation data.** Note that all of the sensitivity and specificity do not change much with respect to the tuning of this parameter. In general Hausdorff distance, as discussed earlier, is very sensitive to outliers or an unexpected magnitude changes in fMRIs. It basically measures the maximum deviation between two signals after they are aligned and since the maximum deviation can get affected by any outlier in the signals, it is very sensitive to such an effect. Precisely to address this issue we introduce the parameter α which provides a relaxation such that our metric does not measure the maximum deviation but instead measures, say, 10^{th} maximum deviation. As a result, it does not get affected by an outlier which is of a shorter duration than 10 time units. However, we do

not get benefits on our simulation data which clearly shows that we do not have any outlier effect. This is, indeed, true since we do not include that in our simulation model. Following is the results for **our real data** for the same case:

	Sensitivity	Specificity
$\alpha = 1$	0.9167	0.5788
$\alpha = 2$	1	0.9182
$\alpha = 5$	1	0.9939
$\alpha = 10$	1	0.9980
$\alpha = 20$	1	0.9667

Table 5.4 Sensitivity and specificity under different α values in real data set

As it is clear from the values in Table 5.4, using α=10 dramatically increases the performance as opposed to using α=1. This actually proves that under real conditions, having outliers is an issue and needs a good care. In our algorithm, with the modification on Hausdorff distance through adjusting the covering ratio, we overcome this issue. Following is two different clustering results for real data with α=2 and α=10 respectively.

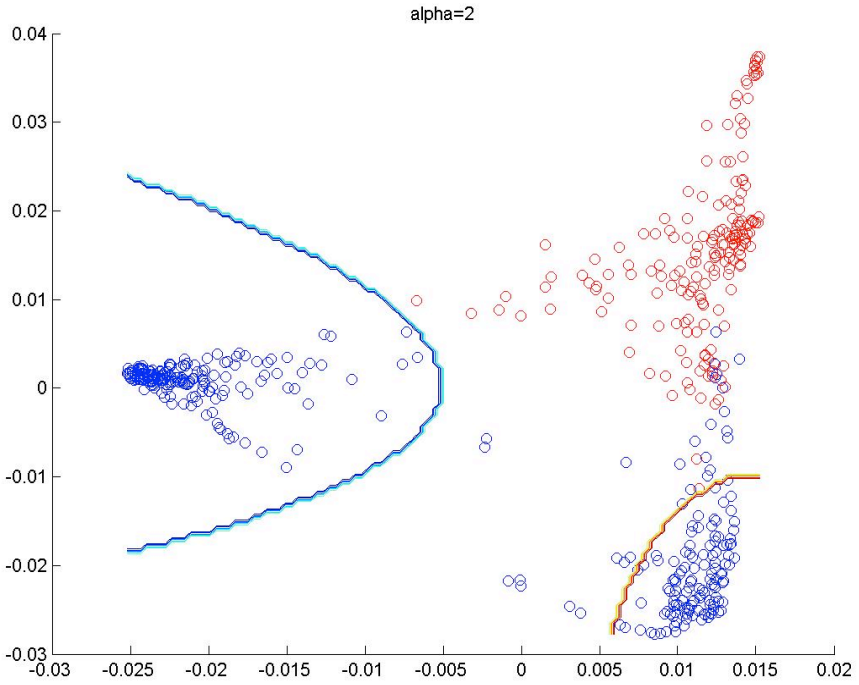


Figure 5.3 Clustering results of real data when α=2

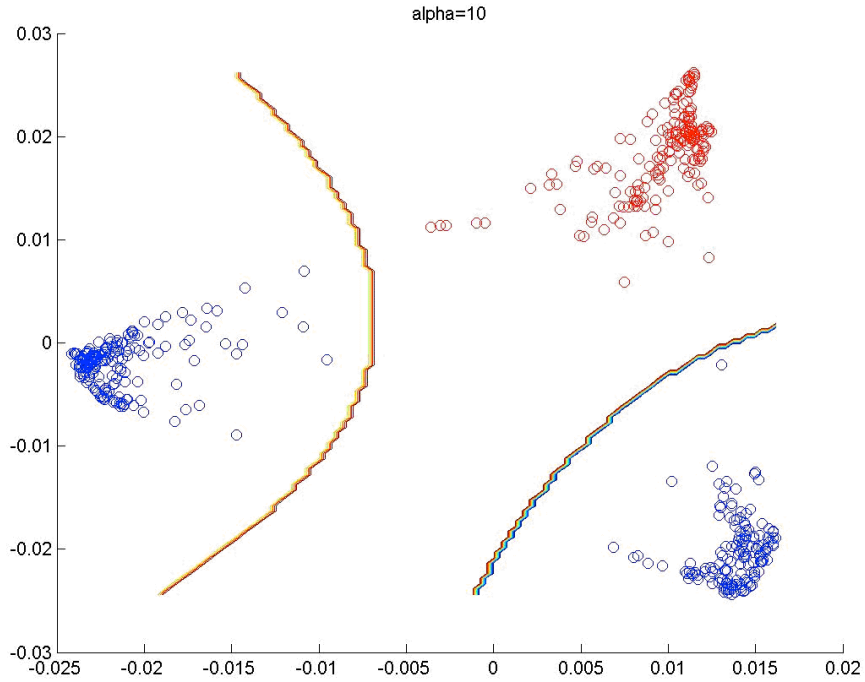


Figure 5.4 Clustering of real data when α=10

As seen in Figure 5.3, when α=0.2 the inactive voxels mix with active voxels and the performance decreases. When we set α=10 as seen in Figure 5.4 the clusters are distinctly separated. So, using a right covering ratio significantly improves the clustering structure in our real data.

Next, we show the effect of the parameter τ on our algorithm together with the following parameter values:

$\kappa = 0.1$
$p = 10$
$\alpha = 10$
$k = 16$

Note that we choose, for now, α as 10, Table 5.5 shows the performance of our algorithm under different choices:

	Sensitivity	**Specificity**
$\tau = 0$	0.9140	0.8500
$\tau = 0.005$	0.9260	0.8560
$\tau = 0.01$	0.9680	0.8520
$\tau = 0.1$	0.9740	0.6720
$\tau = 1$	0.9420	0.6880

Table 5.5 Sensitivity and specificity under different κ values

Recall that our distance metric first aligns two given signals and then measures the α'th maximum deviation. And the parameter τ basically tells how much penalty the algorithm gives to misalignments. It is, so, closely related to the possible lags in the data we work

on. For instance if it is set to 0, then basically we do not give penalty to any lags, in this case and according to our findings, the sensitivity turns out to be low which means that if the active fMRIs gets coupled with passive fMRI's, and if we start not allowing large lags then we get a better sensitivity. Note that sensitivity is maximum when τ is 0.01. If we set it to very large values, then the pairwise distances will dominate and the lag amount starts becoming the only parameter controlling the distribution of the active fMRI's (note that before lags, the pairwise distance for active fMRIs are already very small). This scatters the active fMRIs as a cluster onto a intrinsically one dimensional (only one parameter, lag amount) space, such as a curve. However, this effect would not be realized for passive fMRIs. For this reason even for large settings for this parameter, the sensitivity does not drop much. Figure 5.5 shows our clustering when choosing τ as 1.

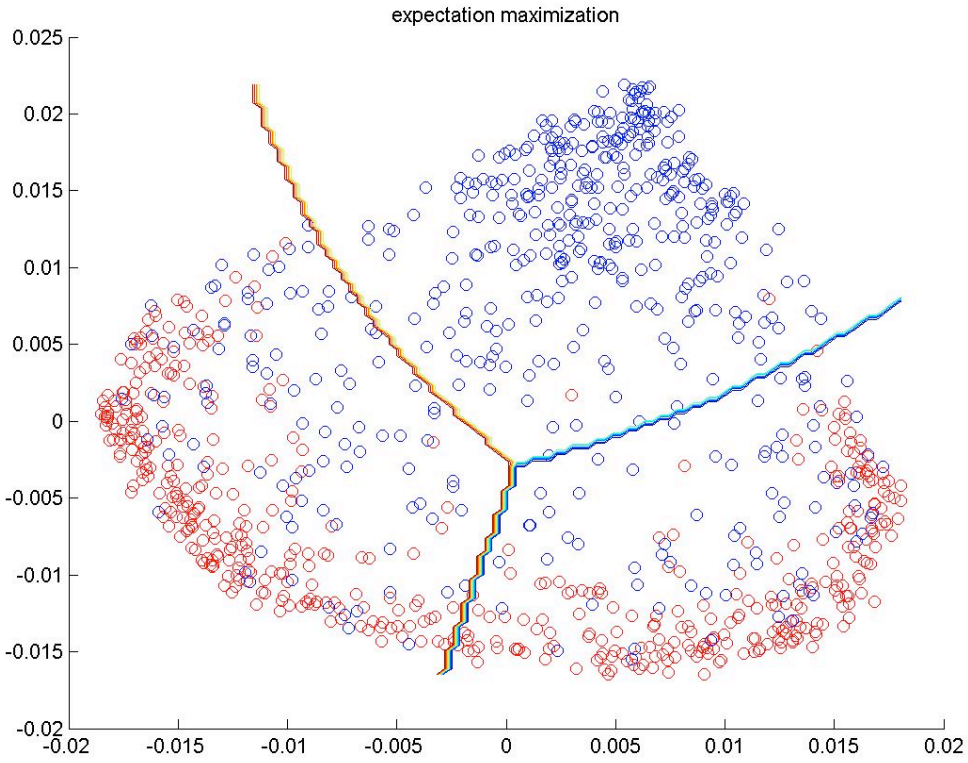


Figure 5.5 Clustering results for simulation data when τ=1

As we discussed, the fMRI's from active class get clustered on an intrinsically one dimensional curve, and on the other hand because of the large variety in the passive class, they are spread out onto the whole spectral domain which decreases the specificity but does not change the sensitivity much.

Next, we show the effect of the parameter *k* on our algorithm together with the following parameter values:

$\kappa = 0.1$
$p = 10$
$\alpha = 10$
$\tau = 0.01$

Note that we choose as τ=0.01, since it gives the greatest sensitivity together with a good specificity. Table 5.6 shows the performance of our algorithm under different choices:

	Sensitivity	**Specificity**
k = 4	0.9140	0.8560
k = 6	0.9700	0.7860
k = 8	0.9600	0.8200
k = 10	0.9560	0.8480
k = 16	0.9680	0.8520
k = 20	0.9780	0.8100

Table 5.6 Sensitivity and specificity under different *k* values

As for the parameter *k* which is used in the spectral clustering, the optimum value for that turns out to be 16. Note that it basically tells how many other fMRI's, an fMRI should be connected to in the graph of Spectral Clustering. For instance, if it is set to 4,

then the algorithm performs poorly. This is because in a neighborhood of an active fMRI of size 4, some passive fMRI's, by chance, can leak in. However, if it is set to a larger value like 16, then most of them should be also active, in turn that it would result in a good clustering. And if it is a too large value, then again an active fMRI would start accepting passive fMRI's in its neighborhood which then would result in a bad clustering. So it should be carefully set and in this case it is better to be 16 according to our findings.

To conclude with, for our simulation data the optimum set of parameters is as follows:

$\kappa = 0.1$
$p = 10$
$\alpha = 10$
$\tau = 0.01$
$k = 16$

Based on our findings, each of these parameters contributes to our fMRI data analysis and clustering differently. Moreover, the results presented in this chapter are in accordance with our original intuition discussed in the theoretical analysis in previous chapters

CHAPTER 6

6 CONCLUSIONS

In this thesis, we conducted fMRI data analysis assuming that there is no extra information about the conducted neural task. Through a blind deconvolution algorithm, we estimated the hemodynamic response function (HRF) within the Bayesian framework using a MAP approach. We showed in our analysis that although this was completely an unsupervised and model free (no particular shape is assumed for HRF) method, we obtained accurate estimates for HRF as well as the unknown stimulus pattern under weak assumptions such as smoothness constraint on HRF. This indicates that even when one has no information about the experimental details of conducted psychological experiment, or when it is not reasonable to use the stimulus pattern (even if it is known) due to high variability in the resulting stimulus perception, it is possible that one can still have a reasonable estimate of not only the response of the brain voxels to the stimulus, but the hemodynamic response function, as well as the stimulus pattern. (In Appendix B we make a comparison with a different method which shows our estimates on the underlying stimulus pattern and the hemodynamic response function are reasonable and successful.)

According to our findings based on our simulations and the real fMRI data, we can also conclude that blind deconvolution together with a smoothness prior on HRF provides a very good approach for fMRI data analysis. Furthermore, posing our methodology within the Bayesian framework using MAP approach is proven to have high efficacy, and brought a new, insightful and an elegant mathematical treatment to the problem.

The only limitation of our approach is that the rise to the peak of an ideal HRF seems to be incompletely reflected in our HRF estimates; there exists an initial offset. In accordance with this, the stimulus is estimated to occur a little earlier in time, and the well known delay between the stimulus exposure on a subject and the brain's response is estimated to be a little smaller. This is basically due to the high degree of freedom in our blind deconvolution. To understand this, we first should note that our estimated HRF with an initial magnitude offset and a less rise to the peak is smoother than the ideal HRF. And one can always find two different patterns of stimulus; one is a delayed version of the other, with which the convolution of those two HRFs (one is the ideal one and the other is the one with an offset) generate the same result. As a result, our blind deconvolution, in this case, favors not the ideal HRF but the one with an initial offset. This type of a limitation is actually a fundamental one, not due to the method itself but the amount of information being exploited. The only way to get around this is just to bring extra information to our Bayesian framework. For instance, the stimulus, if known, can be directly used. However, this is not our initial motivation. Instead one can also assume a lag between impulse and the voxel response. In this case, we could just artificially add the assumed lag in the estimated stimulus, and re-run the E-step of our algorithm once only, to get the corrected hemodynamic response function. Another way, which might be a better one perhaps, is to incorporate the assumed lag in the E-step at each iteration. As said, we see this as a limitation due to the insufficient amount of information assumed to be available to the algorithms as opposed to seeing it due to the blind deconvolution. And possible solutions are available.

In this work, we also studied activation detection among brain voxels through clustering the hemodynamics of the active and inactive voxels. The source of the separation between an active voxel and a passive voxels is reflected in the strong correlation in an active voxel between the stimulus and the fMRI time series. This type of correlation is absent in passive voxels, HRF signals for passive voxels mostly behave randomly. Hence, one can comfortably expect a baseline shape for active HRFs correlated with the stimulus, and no particular shape for passive HRFs. Bearing in our mind that fMRI signals have low SNR, the input to clustering should have the information of underlying stimulus (duration as well as number) and increased SNR. In this respect, using HRF for clustering is not a good idea since it is defined to be the response to a given instantaneous stimulus and ideally we expect it to be of similar shape all over the brain. What makes a voxel passive is not its HRF but the lack of its overall response in our understanding. Hence for clustering, we got the idea of using the response of a voxel not to a stimulus but to the entire pattern of stimulus, which we named in this work as 'hemodynamical time series'. Secondly, since the distance is at the heart of any type of clustering, we investigated Hausdorff distance for fMRI data analysis with spectral clustering followed by an EM algorithm. According to our thorough analysis conducted in the chapter of methodology and results, we can conclude that this approach is robust to AWGN noise, sampling jitter, quadratic drift and lag. In our simulations, even when we consider highly large variance conditions, we generally still have performance above 90% sensitivity and specificity. In particular, we got 100% detection of active and passive voxels for our real fMRI data. Visually speaking, spectrally transformed real data got clustered very nicely and well separated. This proves the capability of Hausdorff distance in terms of capturing the fMRI data distribution. As we analyzed in the methodology, with the modification we used, Hausdorff distance is insensitive to abnormal fMRI magnitude of short durations (which we named outliers) and robust to AWGN noise. It assigns distances mainly considering the maximum magnitude-wise deviation and lag between given signals. Robustness of Hausdorff distance was validated in simulations. This suggests that, in terms of activation detection, fMRI data

distribution has intrinsically 2 dimensions: lag and magnitude. This is our intuition and it requires some further mathematical analysis for justification. Also, with the help of Hausdorff distance, we avoided the burden of feature extraction often performed before clustering since, we think, Hausdorff distance is already highly sensitive to the valid features implicitly. Also based on our comparison with other commonly used metrics, Hausdorff distance turned out to be the one performing best.

To conclude with, we proposed a completely unsupervised and a model-free method for fMRI data analysis which is treated within the Bayesian framework using MAP approach: HRF estimation through MAP blind deconvolution and activation detection through clustering. Based on our findings, our proposed solution proves to be very effective even under variable settings and weak conditions.

Future Work

- In MAP Blind Deconvolution the 'smoothness' constraint may be considered with second order derivatives and also with L_1 norm minimization instead of L_2.
- In MAP Blind Deconvolution we may also put a continuity constraint on the convolution filter (unknown stimulus).
- Hausdorff Distance may be further modified to cover the location correlations of voxels. Since voxels that are closer to each other have similar activations due to the same applied stimulus pattern, these location correlations should be considered while calculating the distances between the hemodynamical signals.
- Experiments with real data sets should be increased.

REFERENCES

[1] Scott A. Huettel, Allen W. Song, Gregory McCarthy, "Functional Magnetic Resonance Imaging", Sinauer Associates, Inc., Sunderland, Massachusetts U.S.A. (2004)

[2] Martin A. Lindquist, Ji Meng Loh, Lauren Y. Atlas, Tor D. Wager, "Modeling the hemodynamic response function in fMRI: Efficiency, bias and mis-modeling", NeuroImage 45 pp. 187–198 (2009)

[3] Cyril Goutte, Finn Arup Nielsen, and Lars Kai Hansen, "Modeling the Haemodynamic Response in fMRI Using Smooth FIR Filters", IEEE Transactions on Medical Imaging, Vol. 19, No. 12, (December 2000)

[4] Worsley, K. J., Friston, K. J. "Analysis of fMRI time-series revisited-again" NeuroImage 2 pp. 173–181. (1995)

[5] Friston, K. J., Penny, W., Phillips, C., Kiebel, S., Hinton, G. and Ashburner, J. "Classical and Bayesian inference in neuroimaging: Theory" NeuroImage 16 pp. 465–483 (2002)

[6] Glover, G. H. "Deconvolution of impulse response in event-related BOLD fMRI" NeuroImage 9 416–429 (1999)

[7] Goutte, C., Nielsen, F. A., Hansen L. K.."Modeling the haemodynamic response in fmri using smooth fir filters" IEEE Trans. Med. Imaging 19 1188–1201 (2000)

[8] Zarahn, E. "Using larger dimensional signal subspaces to increase sensitivity in fmri time series analyses" Hum. Brain Mapp. 17 13–16 (2002)

[9] Riera J. J., Watanabe, J., Kazuki, I., Naoki, M., Aubert, E., Ozaki, T. and Kawashima, R. "A state space model of the hemodynamic approach: Nonlinear filtering of bold signals" NeuroImage 21 547–567 (2004)

[10] Liao, C.,Worsley, K. J., Poline, J.-B., Duncan, G. H. and Evans, A. C. “Estimating the delay of the response in fMRI data” NeuroImage 16 593–606 (2002)

[11] Rajapakse, J.C., Kruggel, F., Maisog, J.M., von Cramon, D.Y., “Modeling hemodynamic response for analysis of functional MRI time-series” Hum. Brain Mapp. 6 (4), 283–300 (1998)

[12] Andersen, A., Gash, D. and Avison, M. J. “Principal component analysis of the dynamic response measured by fmri: A generalized linear systems framework” Magnetic Resonance in Medicine 17 785–815 (1999)

[13] Martin J. McKeown, Scott Makeig, Greg G. Brown, Tzyy-Ping Jung, Sandra S. Kindermann, Anthony J. Bell, and Terrence J. Sejnowski, “Analysis of fMRI Data by Blind Separation Into Independent Spatial Components”, Human Brain Mapping 6:160–188 (1998)

[14] Vazquez, A. L., Noll, D. C. “Nonlinear aspects of the BOLD response in functional MRI” Neuroimage, 7:108-118 (1998)

[15] Stephen LaConte, Stephen Strother, Vladimir Cherkassky, Jon Anderson and Xiaoping Hu, “Support vector machines for temporal classification of block design fMRI data”, NeuroImage 26 317 – 329 (2005)

[16] John-Dylan Haynes, Geraint Rees, “Predicting the Stream of Consciousness from Activity in Human Visual Cortex”, Current Biology, Vol. 15, 1301–1307 (July 26, 2005)

[17] Okito Yamashita, Masa-aki Sato, Taku Yoshioka, Frank Tong, Yukiyasu Kamitani, “Sparse estimation automatically selects voxels relevant for the decoding of fMRI activity patterns”, Neuroimage 42, 1414–1429 (2008)

[18] Yulia Golland, Polina Golland, Shlomo Bentin, Rafael Malach, “Data-driven clustering reveals a fundamental subdivision of the human cortex into two global systems”, Neuropsychologia 46 540–553 (2008)

[19] Goutte, C., Toft, P., Rostrup, E., Nielsen, F., Hansen, L., “On clustering fMRI time series” NeuroImage 9 (3), 298–310 (1999)

[20] Cordes, D., Haughton, V., Carew, J. D., Arfanakis, K., and Maravilla, K. “Hierarchical clustering to measure connectivity in fMRI resting state data” Magn. Reson. Imaging 20, 305–317 (2002)

[21] S. B. Katwal, J. C. Gore, and B. P. Rogers, “Unsupervised Clustering of fMRI Time Series with the Granger Causality Metric” 17th Proc. ISMRM (2009)

[22] C. Davatzikos, K. Ruparel, Y. Fana, D.G. Shena, M. Acharyya, J.W. Loughead, R.C. Gur and D.D. Langleben, “Classifying spatial patterns of brain activity with machine learning methods: Application to lie detection”, Neuroimage 28, 663–668, (2005)

[23] Davatzikos, 2004 C. Davatzikos, “Why voxel-based morphometric analysis should be used with great caution when characterizing group differences” NeuroImage 23, pp. 17–20 (2004)

[24] Mitchell, T. M., Shinkareva, S. V., Carlson, A., Chang, K. M., Malave, V. L., Mason, R. A., and Just, M. A. “Predicting human brain activity associated with the meanings of nouns”, Science 320, 1191–1195, (2008)

[25] K.J. Friston, A.P. Holmes, K.J. Worsley, J.-P. Poline, C.D. Frith, and R.S.J. Frackowiak. “Statistical Parametric Maps in Functional Imaging: A General Linear Approach” Human Brain Mapping 2:189-ZlO (1995)

[26] Friston, K. J., Frith, C. D., Turner, R., and Frackowiak, R. S. J. “Characterizing evoked hemodynamics with fMRI” Neuroimage 2, 157–165 (1995)

[27] V.D. Calhoun, M.C. Stevens, G.D. Pearlson, and K.A. Kiehl “fMRI analysis with the general linear model: removal of latency-induced amplitude bias by incorporation of hemodynamic derivative terms” NeuroImage 22 252– 257 (2004)

[28] Salima Makni, Philippe Ciuciu, “Joint Detection-Estimation of Brain Activity in Functional MRI: A Multichannel Deconvolution Solution”, IEEE Transactions on Signal Processing Vol. 53 No. 9 (September 2005)

[29] Huaien Luo and Sadasivan Puthusserypady, “fMRI Data Analysis With Nonstationary Noise Models: A Bayesian Approach”, IEEE Transactions on Biomedical Engineering Vol. 54 No. 9 (September 2007)

[30] Jorge J. Riera, Jobu Watanabe, Iwata Kazuki, Miura Naoki, Eduardo Aubert, Tohru Ozaki,cand Ryuta Kawashimaa, "A state-space model of the hemodynamic approach: nonlinear filtering of BOLD signals" , J.J. Riera et al. / NeuroImage 21 547–567 (2004)

[31] Roberto Viviani, Georg Gron, and Manfred Spitzer "Functional Principal Component Analysis of fMRI Data", Human Brain Mapping 24:109 –129 (2005)

[32] Martin J. McKeown and Terrence J. Sejnowski, "Independent Component Analysis of fMRI Data: Examining the Assumptions", Human Brain Mapping 6:368–372(1998)

[33] Chad H. Moritz, Victor M. Haughton, Dietmar Cordes, Michelle Quigley, and M. Elizabeth Meyerand, "Whole-brain Functional MR Imaging Activation from a Finger-tapping Task Examined with Independent Component Analysis", AJNR Am J Neuroradiol 21:1629–1635 (October 2000)

[34] Aapo Hyvärinen and Erkki Oja," Independent Component Analysis: Algorithms and Applications", Neural Networks, 13(4-5): 411-430 (2000)

[35] Salima Makni, Philippe Ciuciu, Jérôme Idier, and Jean-Baptiste Poline, "Semi-Blind Deconvolution of Neural Impulse Response in fMRI using a Gibbs Sampling Method" IEEE ICASSP (2004)

[36] Wakako Nakamura, Yujiro Inouye, "Measures of Goodness of Fit to Convolution Model for Analysıs of FMRI Data", 17th European Signal Processing Conference (2009)

[37] Renate Gruner, Bard T. Bjørnara, Gunnar Moen, Torfinn Taxt, "Magnetic Resonance Brain Perfusion Imaging With Voxel-Specific Arterial Input Functions", Journal of Magnetic Resonance Imaging (2006)

[38] D. Srinivasa Rao, K. Selvani Deepthi, K. Moni Sushma Deep, "Application of Blind Deconvolution Algorithm for Image Restoration", International Journal of Engineering Science and Technology (IJEST) Vol. 3 No. 3 (March 2011)

[39] William E. Vanderlinde, James N. Caron, "Blind Deconvolution of SEM Images", Conference Proceedings from the 33rd International Symposium for Testing and Failure Analysis (2007)

[40] James N. Caron, "Efficient blind deconvolution of audio-frequency signal", Research Support Instruments, Quarktet: JASA, Vol 116(1), p. 373-378 (2004)

[41] Selim Esedoglu, "Blind deconvolution of bar code signals", Institute of Physics Publishing, Inverse Problems 20 121–135 (2004)

[42] Laurent Couvreur, "Blind Deconvolution for Multi-Microphone Speech Dereverberation: Application to ASR in reverberant Environments", Proc. 3rd IEEE Benelux Signal Processing Symposium (SPS-2002), Leuven, Belgium, March 21–22 (2002)

[43] Mahdi Karimi, "Rolling Element Bearing Fault Diagnostics using the Blind Deconvolution Technique", PhD thesis Queensland University of Technology, (September 2006)

[44] A. T. Walden "Non-Gaussian reflectivity, entropy and deconvolution", Geophysics, vol. 50, pp. 2862-2888, (1985)

[45] R. A. Wiggins, "Minimum entropy deconvolution", Geoexploration, vol. 16, pp. 21-35, (1978)

[46] Filip Sroubek, Jan Flusser, and Michal Sorel, "Superresolution and blind deconvolution of video", International Conference on Pattern Recognition pp. 1-4 (2008)

[47] Anushri Parsekar, "Blind Deconvolution of Vehicle Inductive Signatures for Travel Time Estimation" Master Thesis Department of Computer Science University of Minnesota Duluth (December 2004)

[48] Axel Baune, Friedrich T. Sommer, Michael Erb, Dirk Wildgruber, Bernd Kardatzki, Gunther Palm, and Wolfgang Grodd, "Dynamical Cluster Analysis of Cortical fMRI Activation" NeuroImage 9, 477–489 (1999)

[49] Francis R. Bach, Michael I. Jordan, "Learning Spectral Clustering", Computer Science University of California Berkeley, CA 94720

[50] Ulrike von Luxburg, "A Tutorial on Spectral Clustering", Statistics and Computing, 17 (4) (2007)

[51] K. Pelckmans, S. Van Vooren, B. Coessens, J.A.K. Suykens, and B. De Moor "Mutual Spectral Clustering: Microarray Experiments Versus Text Corpus" In Proc. of the workshop on Probabilistic Modeling and Machine Learning in Structural and Systems Biology, pp. 55–58 (2006)

[52] Balint Takacs and Simon Butler and Yiannis Demiris, “Multi-Agent Behaviour Segmentation via Spectral Clustering” AAAI-2007 Workshop on Plan, Activity and Intention Recognition (PAIR), pp.74-81 (2007)

[53] Alberto Paccanaro, James A. Casbon and Mansoor A. S. Saqi, “Spectral clustering of protein sequences”, Nucleic Acids Research, Vol. 34, No. 5 1571–1580 (2006)

[54] Habil Zare, Parisa Shooshtari, Arvind Gupta, Ryan R Brinkman, “Data reduction for spectral clustering to analyze high throughput flow cytometry data” Zare et al. BMC Bioinformatics, 11:403 (2010)

[55] William R. Crum, “Spectral Clustering and Label Fusion for 3D Tissue Classification: Sensitivity and Consistency Analysis”, Annals of the BMVA Vol. 2009, No. 6, pp 1–12 (2009)

[56] Boon Thye Thomas Yeo, Wanmei Ou “Clustering fMRI Time Series” December 2, 2004

[57] Daniel P. Huttenlocher, Gregory A. Klanderman, and William J. Rucklidge, “Comparing Images Using the Hausdorff Distance” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, No. 9, (September 1993)

[58] Baofeng Guo, Kin-Man Lam, Wan-Shi Siu and Shuyuan Yang, “Human Face Recognition Using Spatially Weighted Hausdorff Distance”, IEEE International Symposium on Circuits and Systems, vol. 2 pp. 145 - 148 (2001)

[59] Chyuan-Huei ThomasYanga, Shang-Hong Laib, Long-Wen Chang, “Hybrid image matching combining Hausdorff distance with normalized gradient matching” 2006 Pattern Recognition Society. Published by Elsevier Ltd. Pattern Recognition 40 1173 – 1181 (2007)

[60] William J. Rucklidge, “Efficiently Locating Objects Using the Hausdorff Distance” International Journal of Computer Vision 24(3), 251–270 (1997)

[61] Pankaj K. Agarwal, Sariel Har-Peled, Micha Sharir, Yusu Wang “Hausdorff Distance under Translation for Points and Balls” Proc. 19th Annual Sympos. Comput. Geom., pp.282–291 (2003)

[62] Yue Lu, Chew Lim Tan, Weihua Huang, Liying Fan, "An Approach to Word Image Matching Based on Weighted Hausdorff Distance", Sixth International Conference on Document Analysis and Recognition pp. 921-925 (2001)

[63] Young Joon Ahn, "Hausdorff Distance between the Offset Curve of Quadratic Bezier Curve and Its Quadratic Approximation" Commun. Korean Math. Soc. 22 No. 4, pp. 641–648 (2007)

[64] Gabriele Lohmann, D. Yves von Cramon "Automatic labelling of the human cortical surface using sulcal basins", Medical Image Analysis 4 179–188 (2000)

[65] Deepa Kundur, Dimitrios Hatzinakos, "Blind Image Deconvolution" IEEE Signal Processing Magazine 1053-5888/96 (1996)

[66] H.Y.Liu Y.S.Zhang, Song JI, "Study on the Methods of Super-Resolution Image Reconstruction" The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. 37 Part B2. Beijing (2008)

[67] Zheng Lou, "Blind Deconvolution for Image Restoration Using Recursive Filtering" Department of Computer and Electrical Engineering University of Illinois at Urbana-Champaign, (April 2005)

[68] Vazquez, A. L., and Noll, D.C., "Nonlinear aspects of the BOLD response in functional MRI", Neuroimage, 7:108-118 (1998)

[69] Gusnard, D. A., and Raichle, M. E., "Searching for a baseline: Functional imaging and the resting human brain" Nature Reviews Neuroscience 2, 685-694 (October 2001)

[70] C. Sarver E. Rostrup L.K. Hansen Cyril Goutte, F.A. Nielsen "Space-time analysis of fmri by feature space clustering" NeuroImage, pages 7:4, part 2, S610 (1998)

[71] Cyril Goutte, Lars Kai Hansen, Matthew G. Liptrot, Egill Rostrup" Feature Space clustering for fmri meta analysis" Human Brain Mapping, Vol. 13, No. 3. pp. 165-183 (July 2001)

[72] Jueptner M, Weiller C "Review: Does measurement of regional cerebral blood flow reflect synaptic activity? Implications for PET and fMRI" Neuroimage 2:148–156 (1995)

[73] Jagath C. Rajapakse, Frithjof Kruggel, Jose M. Maisog, D. Yves von Cramon "Modeling Hemodynamic Response for Analysis of Functional MRI Time-Series" Human Brain Mapping 6:283–300 (1998)

[74] Larissa Stanberry, Rajesh Nandy, Dietmar Cordes "Cluster Analysis of fMRI Data Using Dendrogram Sharpening" Human Brain Mapping 20:201–219 (2003)

[75] Hesamoddin Jahanian, Gholam-Ali Hossein-Zadeh, Hamid Soltanian-Zadeh, Babak A. Ardekani "Controlling the false positive rate in fuzzy clustering using randomization: application to fMRI activation detection" Magnetic Resonance Imaging 22 631–638 (2004)

[76] Christine Baudelet, Bernard Gallez "Cluster analysis of bold fmri time series in tumors to study the heterogeneity of hemodynamic response to treatment" Magnetic Resonance in Medicine, pages 49:985–990 (2003)

[77] A. Levin, Y. Weiss, F. Durand, W. T. Freeman "Efficient Marginal Likelihood Optimization in Blind Deconvolution" IEEE Conf. on Computer Vision and Pattern Recognition (June 2011)

[78] Boyd, S., Vandenberghe, L. "Convex Optimization" Cambridge University Press (2004).

[79] Coleman, T.F. and Y. Li, "A Reflective Newton Method for Minimizing a Quadratic Function Subject to Bounds on some of the Variables" SIAM Journal on Optimization, Vol. 6, Number 4, pp. 1040-1058 (1996)

[80] Buxton RB, Frank LR. "A Model for the Coupling Between Cerebral Blood Flow and Oxygen Metabolism During Neural Stimulation" Journal of Cerebral Blood Flow and Metabolism, 17:64-72 (1997)

[81] Buxton RB, Wong EC, Frank LR. 'Dynamics of Blood Flow and Oxygenation Changes During Brain Activation: The Balloon Model" Magnetic Resonance in Medicine, 39:855-864 (1998)

[82] Friston KJ, Mechelli A, Turner R, Price CJ. "Nonlinear Responses in fMRI: The Balloon Model, Volterra Kernels, and Other Hemodynamics". NeuroImage, 12:466-477 (1998)

[83] Q. Shan, J. Jia, and A. Agarwala "High-quality motion deblurring from a single image" SIGGRAPH (2008)

[84] S. Cho and S. Lee. “Fast motion deblurring” SIGGRAPH ASIA (2009)

[85] T. S. Cho “Removing motion blur from photographs” PhD thesis, Massachusetts Institute of Technology (2010)

[86] N. Joshi, R. Szeliski, and D. Kriegman “ PSF estimation using sharp edge prediction” In CVPR (2008)

[87] Jiaya Jia “Single image motion deblurring using transparency” In CVPR (2007)

[88] Q. Shan, W. Xiong, and J. Jia “Rotational motion deblurring of a rigid object from a single image” In ICCV (2007)

[89] L. Xu and J. Jia” Two-phase kernel estimation for robust motion deblurring” In ECCV (2010)

[90] Emine Adlı Yılmaz “Wavelet Based Deconvolution Techniques in Identifying fMRI Based Brain Activation” Electrical&Electronics Engineering Department, METU (September 2011)

APPENDICES

APPENDIX A: Preprocessing of fMRI data

Slice-timing correction

Most fMRI data are acquired using two-dimensional pulse sequences to generate thin image planes (slices).The number of slices required to cover the whole brain depends on the capabilities of the scanner. These slices are acquired with equal spacing across the repetition time (TR), but in different orders.

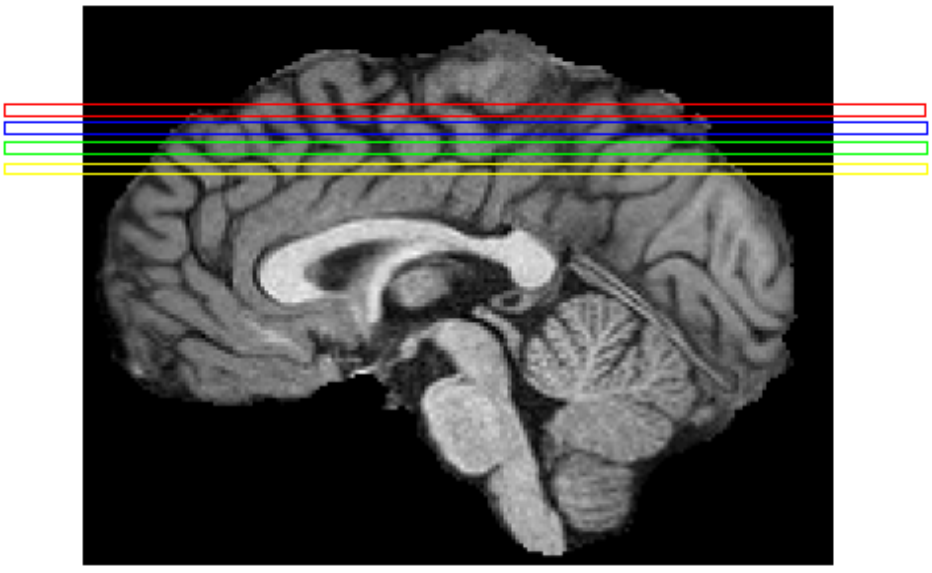


Figure A.1: Slice-timing correction for fMRI data.

Figure A.1 illustrates an example volume with four slices. In order to avoid cross-slice excitation, most pulse sequences use interleaved slice acquisition, in which the odd

slices are scanned first, followed by the even slices. For instance, in Figure A.1, there are four slices in one volume, and each volume is scanned within TR = 3s. The four slices are acquired at 0.75 *s* (red), 1.5*s* (green), A.25*s* (blue) and 3*s* (yellow). However, in data analysis, it is commonly assumed that all these slices in this volume are

acquired at time 0*s*. Such difference in the timing of acquiring each slice is called the slice timing problem. The most commonly used method to correct slice-timing errors is temporal interpolation. In this method, using the information from nearby time points, different interpolation techniques are used to estimate amplitude of the MR signal at the onset of the TR. Thus, for each volume, the intensity of any voxel in that volume is corrected to its intensity values at 0*s*. Although some researchers have proposed more advanced algorithms for slice-timing correction, no method could perfectly recover the missing information from samples. The accuracy of correction depends on the variability in the experimental data and the rate of sampling. Generally, when the variability is low or TR is short, accuracy is higher. For the fMRI data sets with typical temporal variability, slice timing correction is more effective for data acquired at relatively short TRs. For the data sets with longer TRs, slice timing correction could introduce errors. Therefore, this step could be skipped when the TR is long.

Motion Correction

In fMRI analyses, it is assumed that each voxel represents a fixed location of the brain. If the volunteer's head moves, each voxel's time course is derived from more than one brain location. Even small head motion may cause very large damage to raw signal over time. Despite the widespread use of head restraints during fMRI scans, it is hardly possible to keep the head perfectly still. The goal of motion correction is to adjust the time series of images so that $\forall t$, the voxels $v(x, y, z, t)$ in every image correspond to the same position in the brain.

Generally, the process of establishing spatial correspondences between two images is called co-registration. Let M and N be two image volumes. F denotes the spatial transformation that maps voxel coordinates in image M to the coordinates in image N. The coregistration between M and N can be described as an optimization problem:

$$\widehat{\mathcal{F}} = \arg\max_{\mathcal{F}} \left(sim(\mathcal{F}(M), N) + \lambda \cdot R(\mathcal{F}) \right)$$

where $sim(\Im(M), N)$ represents the similarity between the image N and the deformed image $\Im(M) \cdot R(\Im)$ is the regularization on the deformation $\Im$.

Many coregistration methods have been developed for different image modalities. In motion correction, the images of the time series are from the same brain. Therefore, all the volumes in the time series are coregistered to a single reference volume with rigid-body transformation. When using rigid-body transformations for coregistration of two images, it is assumed that the size and shape of the two objects are identical. By a combination of *translations* and *rotations*, one image can be superimposed exactly upon the other.

Here, translation is defined as the movement of the whole image volume along the axes. Let **m** = [x y z]′ be a point in image volume M, where x, y, z are the coordinates in three-dimensional space. The transformation is:

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \alpha_x \\ 0 & 1 & 0 & \alpha_y \\ 0 & 0 & 1 & \alpha_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

where **m** is translated α_x, α_y, α_z units along the axis x, y and z.

Rotation is defined as the turning of the entire image volume around the axes. The Rotation of θ_x radians around axis x is normally described by:

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_x & \sin\theta_x & 0 \\ 0 & -\sin\theta_x & \cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Similarly, rotations around axis y and z can be implemented by the following matrices:

$$\begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \cos\theta_z & \sin\theta_z & 0 & 0 \\ -\sin\theta_z & \cos\theta_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Let $\Omega = \{\alpha_x, \alpha_y, \alpha_z, \theta_x, \theta_y, \theta_z\}$ be the set of parameters in translation and rotation. We denote the rigid-body transformation with parameter Ω on image volume M as $\Im\ (M|\ \Omega)$. The realignment parameters are determined as:

$$\hat{\Omega} = \arg\max_{\Omega} \left(sim(\mathscr{F}(M|\Omega), N) \right)$$

The sum of squared differences or mutual information can be used to measure similarity between the reference and corrected volume. As there is a large number of parameters in Ω, it is challenging to optimize the equation above . Thus, realignment algorithms use iterative approaches for head-motion correction. Gauss-Newton optimization is commonly used in rigid registration.

Spatial normalization

In fMRI analysis, it is sometimes desirable to analyze the functional data from a group of subjects. For instance, some experiments need to examine cross-subject consistency of results. Some researchers try to establish the difference in fMRI responses between healthy and diseased subjects. To analyze fMRI data across subjects, each subject must be transformed into a standard space so that it is the same size and shape as the others. This process is known as spatial normalization, which is an important preprocessing step for most voxel-based fMRI studies. After registration into the standard space, it is generally assumed that the same Euclidean coordinates correspond to approximately the same brain region in all subjects. Although many brain atlases have been proposed and MNI space are the most commonly adopted coordinate systems for spatial normalization.

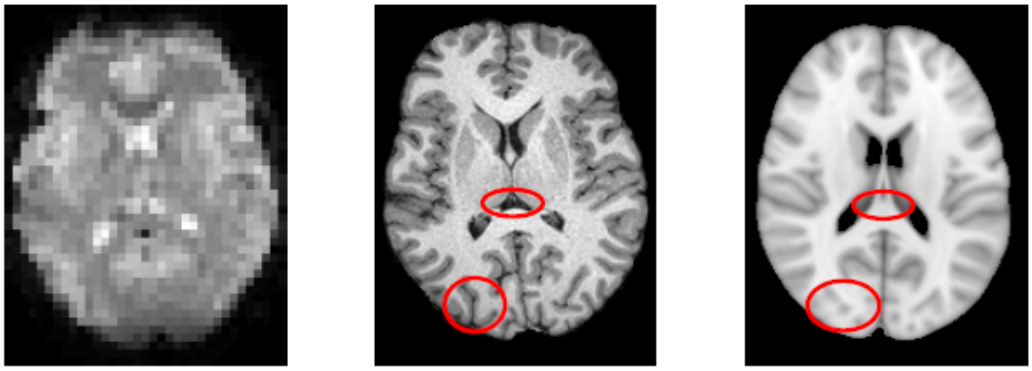


Figure A.2: One slice of functional image, structural image and MNI atlas.

Figure A.2 shows a slice of the functional image (left), the structural image (middle) and the MNI atlas (right). A typical functional image has a relatively low resolution. With this type of image, it is difficult to identify anatomical structures or boundaries and match them with the atlas. On the contrary, high-resolution structural images provide more details. Thus, it is common to acquire a structural image with an fMRI scan. The reference volume of the functional image is first mapped with the structural image using affine registration.

Affine transformations can be described as:

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Let matrix **A** be

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Since most motions for medical imaging applications are reversible, invertibility is a natural requirement for image registration. An affine transformation is invertible if and only if the matrix **A** is invertible. The rigid body transformations introduced previously are a subset of affine transformations. As affine transformations are linear, they can only model the global geometric differences between images. However, as the functional and structural images are acquired with the same brain at almost the same time, affine registration is sufficient to align them with each other. After registering the functional volumes onto the structural image, the structural image is normalized into a standard space. Then, the same transformations are applied to the functional volumes to bring them into the standard space.

APPENDIX B: COMPARISON

In this part we compare and discuss our proposed algorithm (MAP Blind Deconvolution and spectral clustering with EM) with another thesis work [90] in which Fourier Wavelet Regularized Deconvolution (FORWARD) method is used for the purpose of HRF extraction and Laplacian Eigenmaps with fuzzy c-means is used for clustering.

ForWaRD method combines frequency domain deconvolution with frequency domain regularization and wavelet domain regularization. The advantage of deconvolution in the frequency domain is in identifying overlapping signals. But its main weakness is noise amplification. Noise can be reduced in the frequency domain by shrinking frequency coefficients; however, noise and signal may be difficult to separate. ForWaRD solves this by using wavelet domain Wiener shrinkage.

Given a stimulus pattern and an fMRI time series, obtain Fourier transform of *r* (fMRI signal) and *k* (stimulus pattern), called R and K. Convolution corresponds to multiplication in frequency domain. In absence of noise, it is possible to compute an estimate of hemodynamic response function, *d*, through deconvolution shown in (B.1):

$$\widetilde{D}(f_k) := \begin{cases} D(f_k) + \dfrac{N(f_k)}{K(f_k)}, & if\ \left|K(f_k)\right| > 0 \\ 0 \quad , & otherwise \end{cases} \tag{B.1}$$

If noise is amplified at frequencies f_k where $K(f_k)$ is close to zero, parts of noise can appear in the extracted response. ForWaRD uses regularization methods in frequency and wavelet domains in order to overcome this problem. Frequency-domain shrinkage attenuates the noise after the pointwise division, by multiplying each frequency

coefficient $\tilde{D}(f_k)$ by a factor λ(k). λ(k) is the Wiener shrinkage coefficient. After shrinking signal in frequency domain, the leaked noise $D^{-1}n_{\lambda f}$ that Fourier shrinkage λ(k) fails to attenuate has significantly reduced energy in all wavelet coefficients, but the signal part $d_{\lambda f}$ that Fourier shrinkage retains continues to be represented in the wavelet domain. Hence, subsequent wavelet shrinkage effectively extracts the retained signal $d_{\lambda f}$ from the leaked noise $D^{-1}n_{\lambda f}$ and provides a robust estimate.

Differing from our method, ForWaRD estimates HRF for each fMRI signal using the externally applied stimulus. The output of ForWaRD is then used as input to clustering. Before clustering the HRFs, because of the curse of dimensionality problem, a nonlinear dimension reduction method, called Laplacian embedding is performed. Then, fuzzy c-means clustering with cosine distance is used to separate the voxels as active and passive. Figure B.1 shows the comparative HRF extraction results for simulated data with σ_{AWGN}=4 (Figure B.1a). ForWaRD uses the external applied stimulus pattern (red one in Figure B.1c) to estimate the HRF (Figure B.1b). As it is seen it is very similar to our estimation (Figure B.1d) except that ForWaRD catches a deeper undershoot in the estimated hemodynamic.

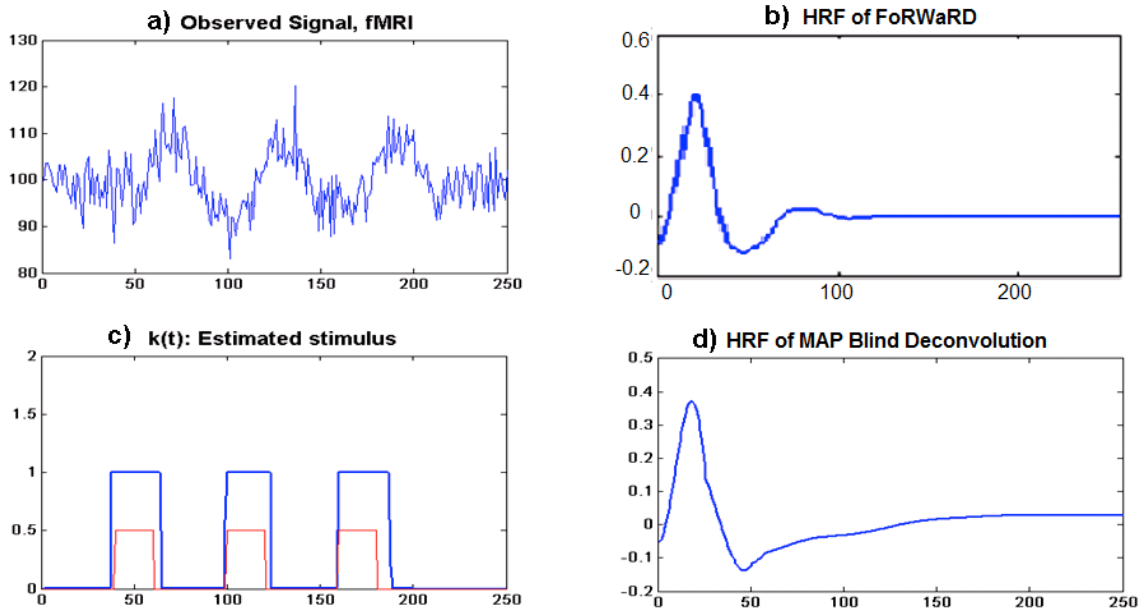


Figure B.1 Comparison of extracted HRFs from simulated data with σ_{AWGN}=4

Figure B.2 shows the results for a real fMRI data from face perception experiment. With the help of our modification for initial rise in MAP Blind Deconvolution, we can successfully estimate the HRF with its rise to peak (Figure B.2d). However, ForWaRD fails to catch this initial rise (Figure B.2b).

This is because the estimated underlying stimulus pattern (blue one in Figure B.2c) is leading the external applied stimulus. It is highly probable that the subject exhibits an anticipation before the stimulus is delivered. Interestingly, as seen in Figure B.2c, at some instants external applied stimulus does not trigger an actual stimulus perception. This is unlikely in a block design, since a cumulative participation is triggered from the underlying neuronal sites. The reason for the absence of stimulus within the blocks need further investigation. However, when ForWaRD uses the estimated stimulus produced via our method, then it can also successfully estimate the HRF recovering the initial rise as shown in Figure B.3b. This is a very good example that our stimulus estimation reflects the actual stimulus as perceived by the subject.

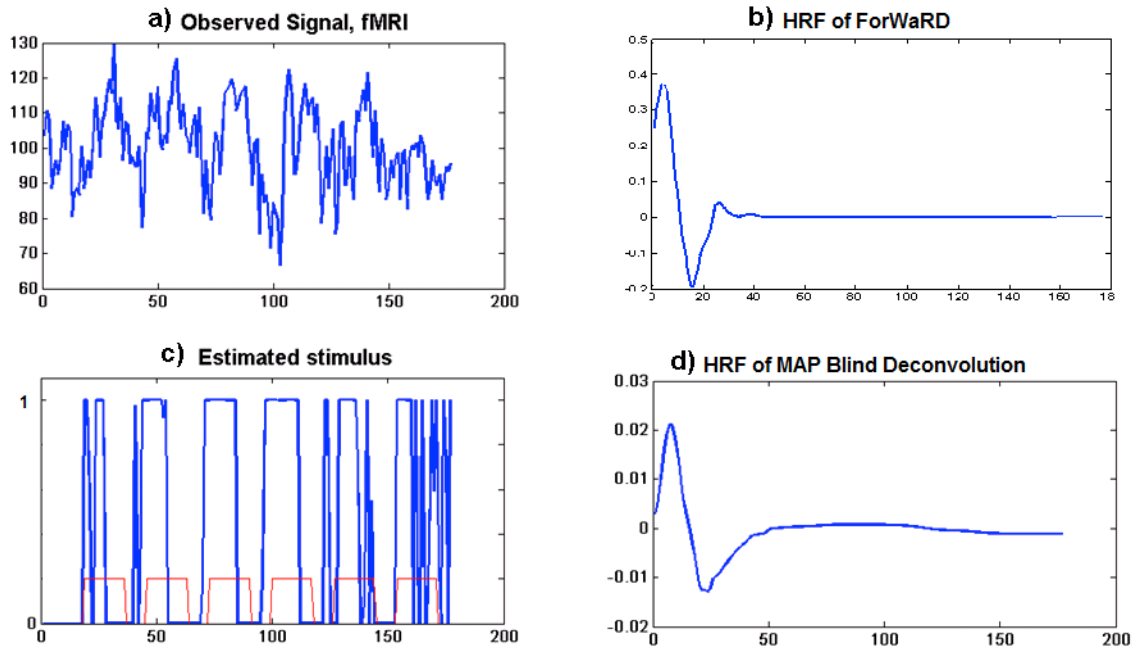


Figure B.2 Comparison of extracted HRFs from real data with voxel ID_100

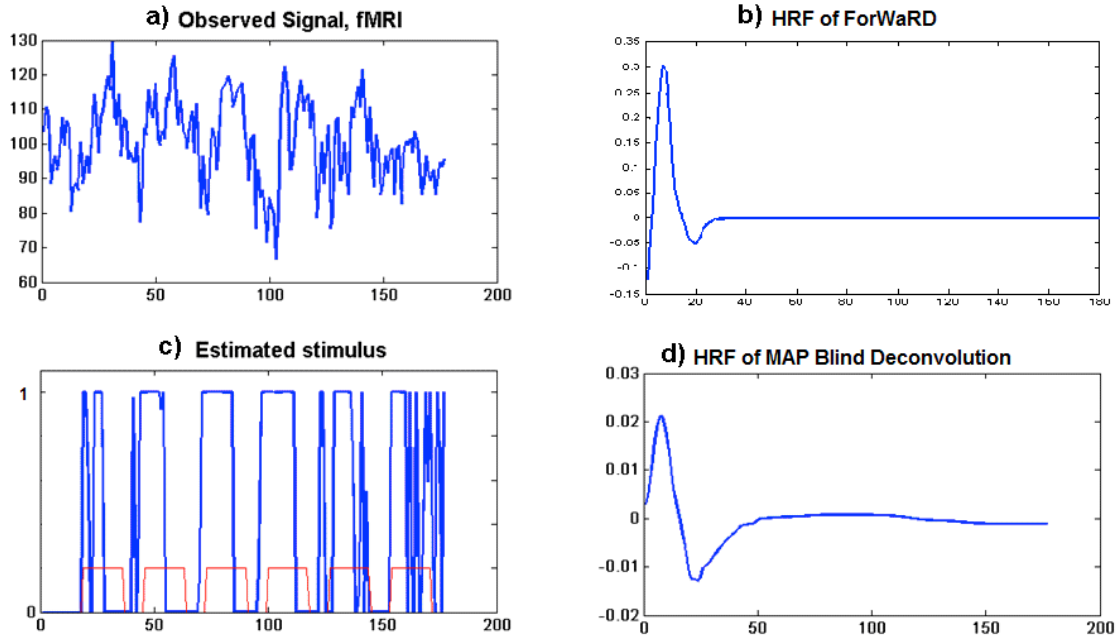
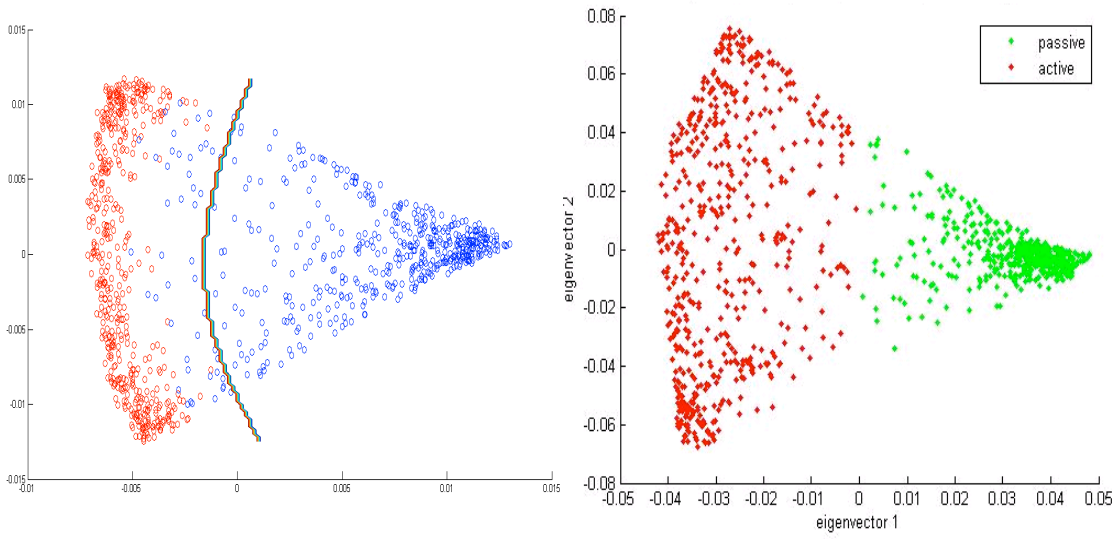


Figure B.3 Comparison of extracted HRFs from real data with voxel ID_100 (ForWaRD uses the MAP estimation of stimulus pattern as input)

To sum up, both methods can successfully estimate the underlying HRF of a voxel. But as we stated earlier, the externally applied stimulus is not always reflected on the subject as it is delivered. The stimulus estimated by our method helps in the recovery of the HRF by both ForWaRD and blind deconvolution methods.

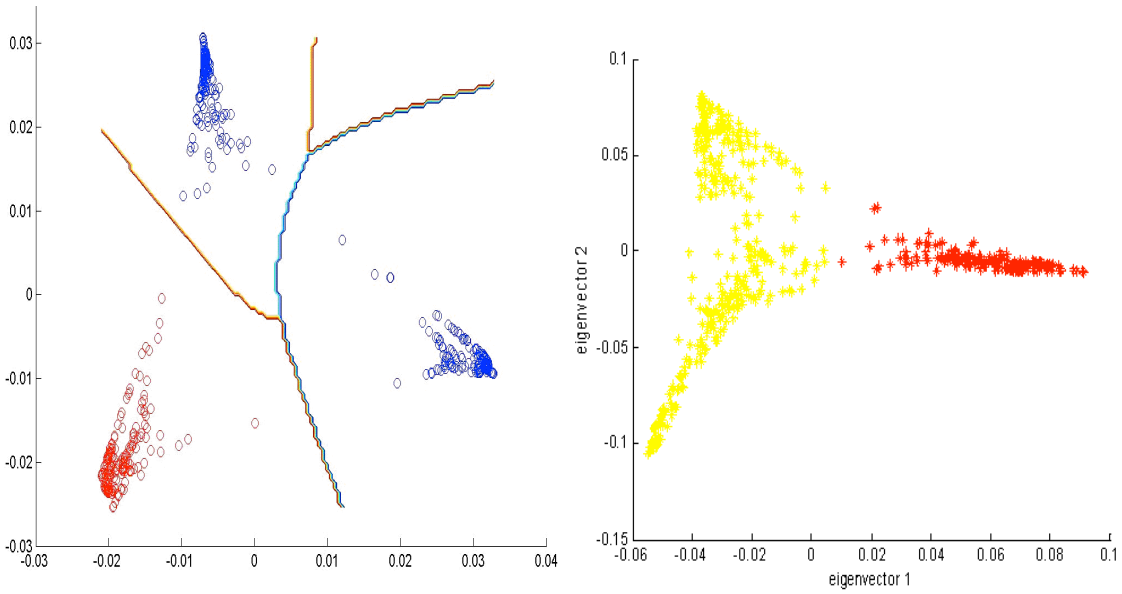
For a comparison of clustering performances we firstly use simulated data with σAWGN=4, σjitter=4, σlag=16, σdrift=16. Figure B.4a shows our results in which the active voxel samples (red) scatter less than the passive ones (blue). In Figure B.4b, using fuzzy c-means clustering with Laplacian Eigenmaps, we observe the opposite case, in which the active data (red) scatter more than the passive ones (green). That means our clustering results are better in sensitivity, while the other method is better in specificity. We get 100% sensitivity, 90.4% specificity and the other method gets 99.8% sensitivity, 92.2% specificity.



a) Spectral clustering with EM *b) Fuzzy c-means with Laplacian Eigenmaps*

Figure B.4 Comparison of clustering results for simulated data

Finally, Figure B.5 shows the clustering results for real data. As shown in Figure B.5a, we get very nice separation among three classes existing in our real data: active, passive and motion. Also the nice nonlinear boundaries provide a natural separation and the cluster centers are distant from each other. On the other hand, in Figure B.5b, although the other method also groups the data into three clusters, the cluster centers are closer than our separation. This is due to the superiority of the Hausdorff distance which can determine similarities and dissimilarities better than the cosine distance that is used in the other method. Here we have 100 % sensitivity and specificity where the other method obtains 98% sensitivity, 99% specificity.



a) Spectral clustering with EM *b) Fuzzy c-means with Laplacian Eigenmaps*

Figure B.5 Comparison of clustering results for real data