Machine Learning Methods for Magnetic Resonance Imaging Analysis

by

Cen Guo

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Statistics)
in The University of Michigan
2012

Doctoral Committee:

Professor Tailen Hsing, Co-Chair
Assistant Professor Long Nguyen, Co-Chair
Professor Douglas C. Noll
Professor Kerby A. Shedden
Professor Naisyin Wang
Associate Research Scientist Scott J. Peltier

ACKNOWLEDGEMENTS

First of all, I would like to express my sincere gratitude to my advisors, Professor Tailen Hsing and Professor XuanLong Nguyen, for their guidance and training throughout my research, especially for their patience and enthusiasm in these years. Without their valuable suggestions and support, this dissertation could not be completed. Besides my advisors, I would also like to thank the rest of the dissertation committee members Professor Kerby Shedden, Professor Naisyin Wang and Dr. Scott Peltier for many insightful comments and questions. I would like to show my gratitude to Professor Tobias Schmidt-Wilcke who brought us the question in the first place and provided insightful knowledge to guide me all the time. Further I would like to thank professor Sawsan As-Sanie from Department of obstetrics and Gynecology, professor Patricia Cagnoli from department of Rheumatology and Dr. Pia Sundgren from department of Radiology for their efforts in gathering and preparing the data. Finally, I would like to thank my parents for their constant support and encouragement during my Ph.D. study and throughout my life.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

ABSTRACT

Machine Learning Methods for Magnetic Resonance Imaging Analysis

by

Cen Guo

Co-Chairs: Tailen Hsing and Long Nguyen

The study of the brain and its connection to human activities has been of interest to scientists for centuries. However, it is only in recent years that medical imaging methods have been developed to allow a visualization of the brain. Magnetic Resonance Imaging (MRI) is such a technique that provides a noninvasive way to view the structure of the brain. Functional MRI (fMRI) is a special type of MRI, measuring the neural activity in human brain. The aim of this dissertation is to apply machine learning methods to functional and anatomical MRI data to study the connection between brain regions and their functions.

The dissertation is divided into two parts. The first part is devoted to the analysis of fMRI. A standard fMRI study produces massive amount of noisy data with strong spatio-temporal correlation. Existing methods include a model-based approach which assumes spatio-temporal independence and a data-driven method which fails to exploit the experimental design. In this work we propose a Gaussian process model to incorporate the temporal correlation through a model-based approach. We validate the method on simulated data and compare the results to other methods through real

data analysis.

The second part covers the analysis of anatomical MRI. Anatomical MRI provides a detailed map of brain structure, especially useful for detecting small anatomical changes as a result of disease process. The goal of anatomical MRI analysis is to train an automated classifier that can identify the patients from healthy controls. We propose a multiple kernel learning classifier which will build classifiers in small regions in the segregating step and then group them in the integrating step. We study the performance of the new method using simulated data and demonstrate the power of our classifier on disease-related data.

CHAPTER I

Introduction

The brain is the most complex organ in the human body with billions of nerve cells. It controls every aspect of our daily lives, such as perception and cognition, movement and regulation, memory and thoughts. For centuries, scientists and philosophers have tried to unravel the complex networks of the brain and its connection to human activities. In the 17th century, people discovered that various areas of the brain had specific functions. Since then understanding the functional regions of the brain becomes a major research area and presents great challenges to the neuroscientists. Before the brain imaging techniques, the studies of the brain function were mainly down by the stimulation of animal brains using electrical currents or the observation of the patients with neurological disorders. However the results showed many inconsistencies and very limited regions could be identified using these methods.

Modern imaging techniques brought a technological breakthrough to the neuroscience, leading to a wave of innovation and enthusiasm in brain studies. These brain imaging methods provide a direct visualization of the structure of the brain, making the studies of living healthy subjects possible. Among them Magnetic Resonance Imaging (MRI) has dominated the neuroscience literature for the current decade because of its high temporal and spatial resolution.

Functional MRI (fMRI) is a special type of MRI. A typical fMRI experiment involves presenting a sequence of stimuli to the subjects while recording the subject's neural activities. It produces a series of scans during one session with temporal resolution varying from 500 ms to 3s. fMRI is particularly useful in cognitive neuroscience research. The fMRI analysis finds the relation between the neural activities and the time course of stimuli. Usually, the main goal of the fMRI analysis is to identify the regions that respond to the stimuli, connecting the regions to the functions.

Structural or anatomical MRI, in general, is used for viewing the structure of the brain. Unlike fMRI, structural MRI acquires only one scan of each subject with high spatial resolution. It provides a good contrast between different tissues, especially

useful for detecting small anatomical changes in the brain. It is known that the neurodegenerative diseases will cause loss of the gray matter which can be discovered by comparing the structural images between the patients and healthy controls. As a result, structural MRI not only becomes popular in brain research but also shows promising results in clinical diagnosis. The goal of the structural MRI analysis is to build a classifier that can distinguish two groups.

Besides brain image's success, it also presents a lot of challenges for the physicists, neuroscientists, psychologists, statisticians, anatomists who involved in the MRI analysis. In the rest of this chapter, we present those issues and discuss different methods to solve them.

1.1 fMRI

fMRI provides a non-invasive way to study the neural activities in human brain with. It works by detecting the changes in blood oxygenation level that occur in response to the local neural activities.

Active neurons consume oxygen. Increases in the local neuronal activities lead to an increase in the local blood flow, carrying more oxygen to the regions with increased activities *Roy and Sherrington* (1890). Oxygen is delivered by haemoglobin in blood cells, which is diamagnetic when oxygenated but paramagnetic when deoxygenated. The small difference in magnetic properties leads to a stronger fMRI signals. Since the blood oxygenation level changes according to the regional neural activities, it can be measured as an indicator of brain activities.

When neuronal activity increases there is an increased demand for oxygen and the local response is an increase in blood flow. This local increase is known as blood oxygenation level dependent (BOLD) signal *Ogawa et al.* (1990). fMRI uses BOLD contrast to study the neural activities in the brain*Huettel et al.* (2009). During a typically fMRI experiment, subjects are asked to perform a certain task while

been scanned repeatedly, giving a series of 3D images. Each voxel in the image is represented by a time series of the signal. Usually the main goal of fMRI analysis is to find the area of the brain activated by the task during the experiment.

The most intuitive solution is to compute the correlation between the recorded signals and the time course of the stimuli and pick the voxels with the highest correlation scores. However brain is a complex network and there are many sources of noises contributing to the signals. The actual analysis is a more sophisticated process than simply computing the correlation scores.

1.1.1 Statistical Parametric Mapping

Statistical parametric mapping (SPM) is a method designed for brain image analysis *Friston et al.* (2007). It builds statistical models to find the regionally specific effects in neuroimaging data, giving a statistical significance map of the investigated regions. SPM is a voxel-based approach which maps all the scans to a template space, reducing any anatomical differences among different subjects. The observations and inferences are made by comparing the same voxels across multiple subjects. In order for the comparison to be valid, all the scans should be mapped into the same space. This is done in the preprocessing steps which include realignment, spatial normalization and spatial smoothing *Friston et al.* (1995a), *Ashburner et al.* (1997), *Friston et al.* (1996a). The preprocessing steps are carried out before the analysis to make the statistical assumptions valid.

General Linear Model

Different statistical analyses of the fMRI are actually different ways to partition the signals into different sources, such as activated signal, confounds and errors according to some assumptions. General linear model is such a method that assumes the signal of interest is a linear function of the haemodynamic response function and

the errors follow an independent Gaussian distributions *Friston et al.* (1995b). There are two concerns about these assumptions. First, the precise mechanism of neuronal activities causing haemodynamic response function is unknown and the shape of the haemodynamic response function may be different across different regions of the brain. Several methods are proposed to model the haemodynamic response function. Second, the errors are not independent for different voxels. Brain images have both strong temporal and spatial correlations which need to be taken into consideration before make any inferences. In general linear model, the temporal correlation is modeled by an autocorrelation model *Woolrich et al.* (2001). The result of general linear model is a map of p-values for the brain regions. However, due to the spatial correlation in the fMRI data, a correction for multiple comparisons is necessary. The theory of random fields provides a way to draw conclusions on those p-values taking the spatial correlation effect into consideration *Worsley et al.* (1996).

1.1.2 Independent Component Analysis

Independent component analysis (ICA) is another way to decompose the fRMI signals (*Calhoun et al.*, 2003). ICA is a dimension reduction technique separating linearly mixed sources into statistical independent components. For fMRI data, it assumes that the observed signals consist of several underlying sources. Calhoun (*Calhoun et al.*, 2003) divided the sources into two groups: signals of interest and signals not of interest where the signals of interest include task-related, function-related and transiently task-related signals and signals not of interest include physiology-related, motion-related and scanner-related signals. All these signals are independent from each other. One advantage of ICA is that it doesn't rely on the connection between neuronal activities and haemodynamic response function. The only assumption ICA needs is that signals are linear mixtures of independent Non-Gaussian components *Hyvärinen and Oja* (2000). And intuitively, the task-related signals should be inde-

pendent from signals not related to tasks, say physiology-related signals. The results of the ICA are brain maps corresponding to each independent component and the activation areas are found by matching the time courses of the components to the design of the experiments. The challenge of the ICA approach is the interpretation of the resulting maps. Unlike the easy interpretation of the parametric map from general linear mode, it is hard to draw convincing conclusions for every component.

1.1.3 Gaussian Process

Gaussian process is a stochastic process that every finite collection of random variables has a multivariate normal distribution. It is widely used to model the temporal and spatial dependent data *Rasmussen and Williams* (2006). The popularity of such processes comes from several reasons (*Davis*, 2001). First, the Gaussian process is completely determined by the its mean and covariance matrix which facilitate the estimation as only the first and second order moments need to be specify. Second, the prediction is easy once given the mean and covariance matrix of the Gaussian process. Third, Gaussian process is a kernel method which is very flexible for various of kinds of correlated data. In this study, we proposed a new method applying the Gaussian process to model the fMRI data. For each voxel, the time series is decomposed into a linear function of the haemodynamic response function, a Gaussian process carrying the temporal dependence information and an independent error terms.

1.2 Structural MRI

MRI (structural MRI or anatomical MRI) uses the phenomena of Nuclear Magnetic Resonance of the nuclei of the hydrogen atom within water. It provides a non-invasive way to visualize the brain. The advantage of MRI over other brain imaging techniques is its superior spatial resolution, providing a detailed map of the brain. Structural MRI has become a powerful tool in both brain research and clinical

neurology. The usual structural MRI experiments scan two groups of different subjects, such as patients vs healthy controls. The main goal of structural MRI studies is to identify the regional changes in the brain that are caused by certain conditions.

1.2.1 Univariate Analysis

The traditional technique of identifying structural changes in the brain is a volumetric measurement method, involving manually drawing regions of interest (ROI) and visually assessing any morphological changes in those regions (*Chan et al.*, 2001), (*Keller and Roberts*, 2009). However, as MRI scans become a standard procedure for both clinical diagnosis and brain research, automated tools are desired to save time and energy from time-consuming manual measurements and subjective assessment. Voxel-based morphometry (VBM) is such a technique proposed by Wright in 1995 (*Wright et al.*, 1995). This method first maps all the scans to a brain template and then constructs a statistical test for every voxel to identify the regional differences between the two groups. It is the counterpart of the GLM in the fMRI analysis and quite successful in distinguishing neurodegenerative diseases (*Whitwell and Jack*, 2005).

Statistical Testing As in the fMRI case, several preprocessing steps are carried out including registration, segmentation and smoothing. After preprocessing step, a statistical test between two group means is applied to every voxel in the image. This involves applying a t-test or a F-test taking any covariates into consideration. The result is a statistical parameter map of the whole brain with a p-value for each voxel. The clusters of voxels with small p-values may be regions that are associated with the disease and need further inspection. Since the statistical parametric map contains the p-values of correlated voxels, multiple test correlation is needed when assessing the significance in any voxels *Friston et al.* (2007).

Application VBM is such an automated method that has been widely used since its first introduction *Ashburner and Friston* (2000). One key reason is that it does not refer explicitly to the brain anatomy and can suit for any MRI analysis. Its application ranges from the studies of brain learning patterns to age-related changes. In particular, it has been successful in characterizing neuroanatomical changes in the brain for various neurodegenerative diseases such as Parkinson's disease (*Price et al.*, 2004), Huntington's disease (*Thieben et al.*, 2002) and Alzheimer's disease (*Karas et al.*, 2003) and mental disorder diseases such as schizophrenia (*Kubicki et al.*, 2002) and bipolar disorders (*Lyoo et al.*, 2004). These works take the VBM approach to identify the significant regions and compare the results to the traditional manual examination method showing that the VBM can detect the regions confirmed by visual assessment method.

Further studies also extend to the healthy subjects, examining the impact of learning and practice on the brain structure. VBM detects the posterior hippocampi region in the brain of the people with extensive navigation experience are significantly larger than the ones of the control group (*Maguire et al.*, 2000). This result is consistent with the idea that the posterior hippocampi region stores a spatial representation of the environment. Another study compares the brain scans of the people before and after learning juggling routine (*Draganski et al.*, 2004). This study shows the expansion in gray matter in bilateral mid-temporal area and left posterior intra-parietal sulcus after the learning process. These regions are shown associated with distance-perception function, visual attention and eye movement. The automatic VBM tool helps to confirm the idea that experience can change the anatomy structure of the brain.

1.2.2 Multivariate Analysis

Although VBM can identify regions that are generally consistent with traditional volumetric method, it does not consider the interrelationship among different voxels and different regions. Recently, machine learning techniques have been playing an increasingly important role in brain image studies. These multivariate techniques are proposed to learn the brain networks. The focus of the new methods shifts from detecting the pathological changes in the brain anatomy to building a classifier that automatically classify the subjects into patient and healthy groups.

Most multivariate methods involve three components (*Fan et al.*, 2007), feature extraction, dimension reduction and classification method. The feature extraction is the key step that determines the quality of the final classifier. One popular feature is the voxel-wise signals of the whole scan as in the VBM (*Asllani et al.*, 2007). The benefit of using the voxel-wise density is that it can achieve the same spatial resolution as the original data. However, there are two problems with this method. One is that the voxel-wise method is very sensitive to the registration error. Another issue lies in the computation efficiency. In order to model the interaction between the voxels, the multivariate methods function in a batch mode, taking the whole scans at one time. Sophisticated machine learning methods can not optimize an objection function with all the voxels in the scan. One solution is to use only the voxels in pre-defined regions (*Cox and Savoy*, 2003). But this method might have selective bias excluding some disease related regions unknown to the scientists before. A better feature will be a one representing the regions other than the voxels. Since the brain images usually have strong spatial correlation and the neighboring voxels share similar values, researchers are more interested in identifying the region effects other than the voxel effects. However, in practice, a prior knowledge about the exact regions is not available. Fan (*Fan et al.*, 2005) computed the correlation between the voxel density and the class label and used it as an indicator of the discriminative power to cluster

the brain into different regions. Tzourlo-Mazoyer provided an anatomical parcellation of the brain through manually drawn regions (*Tzourio-Mazoyer et al.*, 2002).

After defining the regions, one can extract features from each region. In each region, a mixture of Gaussians is applied to model the density function (*Magnin et al.*, 2009). The proposed model is

$$p(x) = \alpha_1 N(x|\mu_1, \sigma_1^2) + \alpha_2 N(x|\mu_2, \sigma_2^2) + \alpha_3 N(x|\mu_3, \sigma_3^2), \tag{1.1}$$

where p(x) is the density function of a region and α_1, α_2 and α_3 are the proportion of CSF, gray matter and white matter in the brain, $\alpha_1 + \alpha_2 + \alpha_3 = 1$. Parameter α_2 representing the gray matter probability is chosen for each region to train a classifier between patients and healthy controls. The benefit of using region-based features is that it is very robust. By summarizing a few features to represent the regions, it reduces the effect of noise from preprocessing steps and individual variation.

Different machine learning methods have been proposed to classify the two groups. Robin Wolz (*Wolz et al.*, 2011) compared linear discriminant analysis method with support vector machine. The results showed that linear discriminant had better specificity while support vector achieved better sensitivity. Phillips (*Phillips et al.*, 2011) applied relevance vector machine to vegetative state patients. Deanna Greenstein (*Greenstein et al.*, 2012) used a random forest algorithm to the children with onset schizophrenia. The accuracy of those classifiers largely depends on the extracted features in the previous step.

1.2.3 SVM and multiple kernel analysis SVM

Support vector machine (SVM) proposed by Vapnik (*Vapnik*, 1995) is a kernel-based classification method which achieves great success especially in high-dimension problem. Several reasons lead to its popularity. First, the formulation of SVM is

intuitive and easy to understand. Second it is a kernel-based method which is flexible with a broad range of problems. Third it suits small sample and high dimension problems well. As in the VBM case, SVM was used in the function MRI to predict the state of the scans during a block design experiment (*LaConte et al.*, 2005). Then it was proposed for structural MRI, achieving good results in various kinds of data. Lao (*Lao et al.*, 2004) first applied the SVM to the structural MRI to determine the gender of the subjects. The study showed that SVM could easily distinguish the two groups, achieving a classification accuracy of 97%. Kawasaki (*Kawasaki et al.*, 2007) applied SVM to classify the schizophrenia patients from the healthy controls. Klöppel (*Klöppel et al.*, 2008) successfully distinguished the Alzheimer's patients from normal people with an accuracy of 89%.

The performance of SVM relies on the kernel which is determined before seeing the data. Selecting a kernel and its parameters is an important issue in training. The classical way is to use cross-validation procedure which requires an extra validation set. However, in a small sample problem, extra data are usually hard to acquire. Multiple kernel learning (MKL) is proposed to automatically select the best kernel. It takes a weighted sum of different kernels instead of using a single one (*Lanckriet et al.*, 2004), (*Sonnenburg et al.*, 2006). Since the weight of each kernel is automatically determined by the MKL algorithm, it does not need extra data to select the best kernels. There are two uses of MKL (*Gönen and Alpaydin*, 2011). First one is to get a kernel as a combination of pre-defined kernels. Different kernels correspond to the similarity between two subjects in different spaces and MKL finds the best combination of all these spaces instead of just picking one. Second one is to get a kernel as a combination of different sources. Different variables can have different measures and can be best represented through different features. In such a case designing different kernels for different variables and combining the kernels later are a way of using multiple information sources. This means that different variables may

contribute to the classifier in different ways. This is intermediate combination (combining kernels taken different data), different from early combination (combining the data at feature extracting step, single kernel SVM) and late combination (combining different classifiers taken different data) (*Noble*, 2004).

The latter usage can be used in the classification problem of the brain image data. Human brain exhibits both segregation and integration properties (*Kinser and Grobstein* (2000)). Segregation means that different aspects of the behaviors are usually performed by anatomically and functionally distinct areas. Integration means that these functionally specialized areas need to communicate with each other to complete any tasks. These localization and globalization property of the brain can fit in the framework of MKL method which extracts information from different local areas and then combines them together to get a better result. In this work, we present and evaluate a classification method based on MKL SVM. The purpose is to distinguish the patients with a certain disease from the healthy control subjects through the analysis of their anatomical brain images. In addition, we are also interested in the identification of significant regions associated with a particular disease.

1.3 Overview

The material presented in this thesis covers both the fMRI analysis and structural MRI analysis, including theoretical and practical backgrounds, simulation and real data analysis.

Chapter II devotes to the fMRI analysis. Section 2.1 listed some characters of fMRI data and define the purpose of fMRI studies. We also presented several challenges fMRI data bring to the statistical analysis in this section. Section 2.3 reviews the general linear method with its ways of modeling the haemodynamic response function and dealing with temporal and spatial correlation in the data. In section 2.4 we first explain the intuition of ICA and then dig into the details of its algorithm

and application. Section 2.5 gives a description of Gaussian process and the decomposition model that we proposed to model the temporal dependent data. We applied the proposed method to a simulated data and the results show that the estimates of the parameters are stable around the truth. In section 2.6, we apply our model to an auditory stimulation data set and compare our results to GLM and ICA.

Chapter III covers the analysis of structural MRI. In section 3.2, we describe the voxel-based method, along with its preprocessing steps and statistical tests. In section 3.3, we focus on the mathematical formulation of MKL SVM. We introduce both the primal and dual forms of the problem which can show the benefits of these segregating and integrating procedures. In section 3.4, we compare the results of traditional SVM and MKL SVM in simulated data. The simulation results show that MKL SVM can achieve a better classifier and identify the informative variables from the noise variables. In the section 3.5, We propose a two-step MKL procedure to deal with highly correlated areas in the brain. We apply the method to four data sets, showing that the MKL SVM can outperform the traditional SVM in some conditions. We also discuss some common feature selection and tuning selection issues in the real data analysis.

CHAPTER II

Functional MRI Analysis

2.1 Introduction

Functional Magnetic Resonance Imaging (fMRI) provides a non-invasive way to study neural activity in human brain. It is known when local nerve cells are active, there is an increase of local blood flow after an approximately 1-5 second delay (*Roy and Sherrington*, 1890). This leads to local changes in blood oxygen level which are reflected as an increase of magnetic resonance signals (*Ogawa et al.*, 1990). fMRI uses this blood oxygenation level dependent (BOLD) contrast to study local neural activity in the brain. During a typically fMRI experiment, subjects are asked to perform a certain task while been scanned repeatedly. Each scan is a 3-D image of the whole or a part of the brain. A typical fMRI scan has a spatial resolution of about several millimeters (usually $3 \times 3 \times 3$ mm^3) and time resolution of about a few seconds (usually 2 seconds). It is known that different areas of brain are associated with different functions, such as analyzing sensory data, performing memory functions and making decisions. The goal of activation analysis is to find the activated area of the brain when the subject is performing a certain task.

The most widely used statistical method is the general linear model (*Friston et al.*, 1995b) which builds a linear model between the recorded signal and the expected activation signal. Since the local blood flow usually does not synchronize with the stimuli, the expected activation signal is represented by a convolution model between the time course of the stimuli and the HRF. The general linear model method assumes the observed signal is a linear function of the expected signal plus some random noise. Then a statistical test is applied to every voxel to test whether the linear association is significant or not. The significant regions are the active areas invoked by the stimuli.

The statistical analysis of fMRI data is challenging due to several reasons. First, the precise mechanism linking BOLD signal and neural activity is not clear, which means the shape of HRF is not known. Second a standard fMRI study produces massive amounts of data, with strong temporal and spatial correlation. Further, the

signal we are interested in is weak in comparison to the noise which can come from several sources, like head movement, equipment, effect of respiration and heartbeat. The signal intensity in a given voxel typically varies by approximately 5% around the mean during cortical task activation.

One solution to avoid the use of HRF is independent component analysis (*Bell and Sejnowski*, 1995) which decomposes the recorded signal into several independent components. This method does not assume any relation between the observed signal and the activation signal in the model estimation step. Instead, it assumes the recorded signals are a linear combination of independent components and the goal is to retrieve the original sources. The components with high correlations with the stimuli are the activation sources. And the regions relies mostly on the activation sources are the active regions invoked by the stimuli. The difficulty with this method lies in the interpretation of the independent components and the activation regions.

In general linear model, the temporal correlation of the signal is usually modeled by an autoregressive model for every voxel. The order of the autoregressive is fixed for all the voxels across the brain. However, the strength of the temporal correlation varies for different regions. Gaussian process is a good method to model the stochastic process with temporal correlation. Gaussian process is a stochastic process for which any collection of finite variables follow a multivariate Gaussian distribution (*Rasmussen and Williams*, 2006). One advantage of Gaussian process is its flexibility in designing the level and the structure of the correlation through the covariance matrix. We explore this advantage and design a Gaussian process model to model the fMRI signal.

The rest of this chapter provides more details of statistical methods on fMRI data. We first describe the preprocessing steps that are part of standard procedure now. We review two popular methods, general linear model and independent component analysis. Then we propose a new decomposition method using Gaussian process. We

show that our model can recover the activation signal, the temporal correlated signal and the random noise in a simulated study. We then compare our model to other methods in a real data analysis.

2.2 Preprocessing

Most brain image analyses are voxel-based, relying on the voxel-wise signals to find the regional effects in the data. This requires all the scans to be in the same space with same voxel indicating same location across multiple scans. To meet this requirement, a series of preprocessing steps, including realignment, spatial normalization and spatial smoothing, are usually carried out before the analysis to make the statistical assumptions valid.

Realignment Signal changes in one voxel of one session can arise from head motion of the subject. In extreme cases, the movement can account for up to 90% of the total noise (*Friston et al.*, 1996b). A typical fMRI scan has a spatial resolution of $2mm$ but the subjects usually show displacements of up to several millimeters. The time series of voxel i may be contaminated by voxel j, a few millimeters away. So before dealing with the variability between different sessions and different subjects, a realignment procedure is applied to all the scans to eliminate the within session variability. The Realignment involves a rigid-body transformation, minimizing the differences between each successive scan and a reference scan. Then the transformation is applied to each scan and a re-sampling scheme is carried out to get the signal on the grid. The results are a series of scans aligned to the same space. Sometimes, non-linear transformation is applied to account for non-linear effects.

Spatial Normalization Realignment reduces the within-session differences among a series of scans. But different subjects have different brain morphometries which must be taken into consideration before statistical analysis. Spatial normalization is the step that maps all the scans to a same template image (*Friston et al.*, 1995a). After the realignment step, a mean image of the series or a structural image is used to estimated the warping function that maps it onto a template (*Talairach and Tournoux*,

1988). In practice, most people use a spatial basis function to minimize the difference between the two images. After spatial normalization step, all the scans in the study are in the same space in which voxels lie in the same location across different scans.

Spatial Smoothing After the spatial normalization steps, images are usually smoothed by convolving with a Gaussian kernel. For fMRI data, the signal of interests is usually very weak comparing to the noise. Smoothing reduces the noise, improving the signal to noise ratio. Another reason is that smoothing makes the errors more Gaussian, an assumption in the statistical analysis.

2.3 General Linear Model

Friston et al. (1995b) used the general linear model (GLM) for activation analysis. In their work the time series of each individual voxel is modeled independently as a linear combination of the experiment-related signals and white noise. Let $X(t)$ be the time series at any voxel, then

$$X(t) = \beta_0 + \beta_1 g_1(t) + \beta_2 g_2(t) + \ldots + \beta_K g_K(t) + \epsilon(t), \tag{2.1}$$

where g_k is the explanatory variable relating to the k-th experiment condition. Putting equation (2.1) in matrix form

$$\mathbf{X} = \mathbf{G}\beta + \epsilon, \tag{2.2}$$

where $\mathbf{X}$ is the observed time series at a specific voxel. Matrix $\mathbf{G}$ is the design matrix with columns g_k. ϵ is the white noise $\epsilon \sim \mathcal{N}(0, \sigma^2\mathbb{I})$.

The covariate $g_k(t)$ is the expected BOLD signal corresponding to the k-th experimental condition. The BOLD signal evoked by a single stimulus is referred to as the haemodynamic response function (HRF). Due to the sluggish nature of the HRF, the BOLD signal is not a linear function of stimulus function. A common way to model

BOLD signal is through a linear time-invariant model.

Let $u_k(t)$ be the time series of stimulus of condition k and $h(t)$ be the HRF, linear time-invariant model expresses BOLD signal as the convolution of the stimulus function and HRF (*Boynton et al.*, 1996).

$$g_k(t) = u_k(t) \otimes h(t) = \int u_k(t-\tau)h(\tau)d\tau. \tag{2.3}$$

There are usually two types of experimental design, epoch and event-related, which lead to different expressions of function $u(t)$. In epoch model, $u_k(t)$ is a boxcar function, with value 1 at the time when condition k is on and 0 otherwise,

$$u(t) = \sum_{j=2}^{J} I_{(t_{j-1},t_j)}(t),$$

where $I(t)$ is an indicator function and (t_{j-1}, t_j) is the j-th block when the stimulus is on. Although block design is efficient in detecting the activated area, it only measures the magnitude of BOLD signal. In event-related design, $u_k(t)$ is a stick function (*Zarahn et al.*, 1997):

$$u(t) = \sum_{j=1}^{J} \delta(t - t_j), \tag{2.4}$$

where $(t_1, \ldots, t_J)$ is the time series of J stimuli and $\delta(t)$ is the Dirac delta function. Event-related design is usually used to characterize transient haemodynamic responses to brief stimuli (*Josephs and Henson*, 1999). It facilitates an evaluation of the exact form of the HRF.

GLM is a massive univariate model which involves two-step analysis. First linear model (2.2) is applied to each time series separately. Then a test statistic (usually T-statistic or F-statistic) of the null hypothesis that there is no activation for condition

k is computed for each voxel,

$$H_0 : \beta_k = 0 \qquad H_1 : \beta_k > 0.$$

This creates a statistical image with each voxel represented by its corresponding test statistic. At the second level of analysis, a threshold for test statistic is chosen based on some multiple testing correction techniques such as random field theory (RFT) and false discovery rate method (FDR).

GLM is the dominant method to analyze fMRI data mainly due to its computational simplicity. There are several issues about this method. First it fails to accommodate the spatio-temporal correlation in the data. The temporal correlation is due to the sluggish nature of BOLD. The spatial correlation can come from several sources, like image reconstruction and preprocessing steps. Second, it performs a extremely large number of tests simultaneously (usually on the scale of 100000) with multiple comparison issues. Third, the power of GLM strongly depends on the form of the HRF. It has been observed that the exact form of the HRF varies across different regions of the brain. Fixing the form of the HRF largely reduces the flexibility of the model. New methods have been proposed to deal with above issues.

2.3.1 HRF

The mechanism between neural activity and BOLD signal is complicated and only partially understood. A typical HRF, the BOLD response to a single stimulus, usually peaks approximately 5s after stimulation, and is followed by an undershoot that lasts as long as 30s, as showed in Figure 2.1.

This canonical HRF is widely used as a basis function in (2.3). Empirical studies show that the shape of the HRF is similar across sensory regions in brain, for example motor cortex (*Aguirre et al.*, 1998), visual cortex (*Boynton et al.*, 1996) and auditory

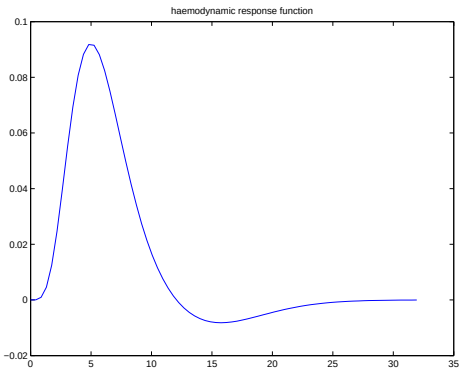

Figure 2.1: Canonical hemodynamic response function $h(t)$

cortex (*Josephs et al.*, 1997). However, the precise shape of the HRF is unknown and still an active research area. Parametric methods focus on modeling the characters of the HRF such as amplitude, onset latency, peak latency and dispersion (*Rajapakse et al.*, 1998).

Basis Function

The precise shape of the HRF is different in different regions in the brain. Using one canonical basis function can not accommodate this variability. One way to increase the flexibility of the model is to expand to I basis functions, $f_1(\tau), \ldots, f_I(\tau)$, and express the HRF as a linear combination of these K basis functions, $h(\tau) = \sum_{i=1}^{I} \gamma_i f_i(\tau)$. In this case the BOLD signal evoked by a single condition is

$$g(t) = u(t) \otimes \sum_{i=1}^{I} \gamma_i f_i(\tau) = \sum_{i=1}^{I} \sum_{j=1}^{J} \gamma_i f_i(t - t_j),$$

where $u(t)$ is an event-related design and takes the form of (2.4). Having specified the form of stimulus and the HRF, equation (2.1) can be written as

$$\begin{aligned} X(t) &= \sum_{k=1}^{K} \beta_k g_k(t) + \epsilon(t) \\ &= \sum_{k=1}^{K} \beta_k \sum_{i=1}^{I} \sum_{j=1}^{J} \gamma_i f_i(t - t_{kj}) + \epsilon(t) \\ &= \sum_{k=1}^{K} \sum_{i=1}^{I} \sum_{j=1}^{J} \alpha_{ki} f_i(t - t_{kj}) + \epsilon(t), \end{aligned}$$

where $\alpha_{ki} = \beta_k \gamma_i$ and $(t_{k1}, \ldots, t_{kJ})$ is the time series of stimulus function for the k-th condition. A common choice for function $(f_1(t), \ldots, f_I(t))$ is based on the canonical HRF and its partial derivatives. The canonical HRF can be characterized by the difference between two gamma functions, one modeling the peak and the other modeling the undershoot. Derivative function can capture the difference in onset latency among different brain regions.

Another popular set of basis functions is three gamma density functions with mean and variance both setting to 2^i ($i = 2, 3$ and 4). They can be seen as functions peaking during the early, intermediate and late components of the anticipated haemodynamic response. And also derivatives of these three basis functions are used in the case when there is temporal delay effect. Figure 2.2 plots the basis functions.

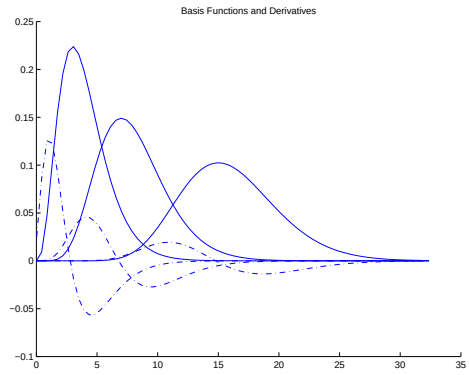

Figure 2.2: Basis function $f_k(t)$ and its derivatives

2.3.2 Temporal Correlation

Purdon and Weisskoff (1998) investigated the temporal correlation in fMRI data. A simulation study demonstrated that the false-positive rate can be biased far above or below the significant level if the actual autocorrelation was ignored. Instead of assuming independence along the time series, they proposed a new method using first-order autoregressive (**AR**(1)) model to accommodate the temporal correlation in noise term. The error term $\epsilon(t)$ in (2.1) is modeled as a sum of **AR**(1) series and white noise,

$$\begin{aligned} \epsilon(t) &= z(t) + \delta_\epsilon(t) \\ z(t) &= az(t-1) + \delta_z(t), \end{aligned}$$

where $\delta_\epsilon(t)$ and $\delta_z(t)$ are independent Gaussian, $\delta_\epsilon(t) \sim \mathcal{N}(0, \sigma_\epsilon^2)$, $\delta_z(t) \sim \mathcal{N}(0, \sigma_z^2)$. a is the **AR**(1) coefficient. Then the resulting covariance matrix is

$$\mathbf{E}(\epsilon\epsilon^T) = \sigma_z^2(\mathbb{I} - A)^{-1}(\mathbb{I} - A)^{-T} + \sigma_\epsilon^2\mathbb{I},$$

where A is a matrix with all elements of the first lower off-diagonal set to a and zero elsewhere. $\mathbb{I}$ is the identity matrix of dimension T.

2.3.3 Multiple Testing Correction

Correction for multiple testing is crucial for the interpretation of activation analysis. A typical fMRI experiment produces massive amounts of voxels with strong spatial correlations. The reasons for spatial correlation come from different sources, such as image reconstruction, physiological signal and spatial preprocessing. Since the BOLD signal is relatively low comparing to noise, the standard preprocessing step involves smoothing along the spatial direction, usually with a Gaussian kernel of full

width at half maximum (FWHM) of 8 pixels. At the modeling level, GLM assumes independence among voxels and fits each time series individually. This spatial correlation is addressed in inference step through multiple testing correction technique.

Bonferroni Correction

There are several methods to address the multiple comparison issue. One way is to control the family-wise error rate ($\mathrm{P}^{\mathrm{FWE}}$) using Bonferroni correction. The significant level α for an individual test is then

$$\alpha = \mathrm{P}^{\mathrm{FWE}}/N,$$

where N is the number of individual tests (number of voxels in brain). However, in standard fMRI experiment, we deal with about 100000 multiple tests simultaneously. The Bonferroni correction is too conservative. Further scans have spatial correlation which makes the effective degree of test statistics much smaller. Usually Bonferroni correction does not lead to correct family-wise error rate.

Random Field Theory

This spatial dependence problem can be corrected using random field theory (RFT) (**?**). The way that RFT solves this problem is through expected value of *Euler Characteristic* (EC) for a smooth statistical map. *Euler Characteristic* is defined as the number of clusters of voxels that exceed a given threshold in the brain volume (*Worsley et al.*, 1996).

False positive rate is the probability of at least one voxel activated which is equivalent to the largest Z value in one region is above some threshold. This is the same as the probability of finding at least one region above the threshold. And at high thresholds the EC is either is zero or one. so we have the probability of a family-wise

error is approximately equivalent to the expected Euler characteristic:

$$\mathrm{P}^{\mathrm{FWE}} = \mathbf{P}(Z_{max} > \mathrm{Z}_\alpha) = \mathbf{P}(\mathrm{EC} \geq 1) \leq \mathbf{E}(\mathrm{EC}).$$

False Discovery Rate

Instead of controlling type I error, false discovery rate (FDR) approach tries to control the expected proportion of false positives among those tests detected as positive (*Benjamini and Hochberg*, 1995). The algorithm is to calculate the p-value for each individual voxels and order them, $p_1 \leq p_2 \leq \ldots \leq p_N$. To control FDR at level α, the largest k which satisfies $p_k < \alpha k/N$ was found. Then tests associated with $p_1, \ldots, p_k$ are considered as positive. FDR approach shows higher power than Bonferroni correction in fMRI data set (*Genovese et al.*, 2002). The resulting threshold chosen by FDR depends on the amount of significant signals in the data set not on the number of voxels or the smoothness in the data. So unlike single choice of threshold across data sets, FDR method adapts its threshold to the features of the data. On the other hand, ignoring the smoothness in the data sets, FDR tends to be more conservative as the spatial correlation increases. Hence, it has higher power for unsmoothed data while RFT typically has higher power for smoothed data.

2.4 Independent Component Analysis

GLM requires a priori knowledge about the exact form of the HRF. In the brain regions where the HRF is quite different from the canonical form, GLM can not detect the activation area. *McKeown et al.* (1998) proposed a novel approach which does not specify the shape of the HRF. The method is based on independent component analysis (ICA) (*Bell and Sejnowski*, 1995). Like PCA, ICA decomposes a time series of scans into a linear combination of several sources and associated weights. But unlike PCA which tries to find the best solution in terms of minimizing the mean-

square error, ICA decomposes the original signals into components as independent as possible.

2.4.1 Definition of ICA

Let $X_1, \ldots, X_J$ be J random variables. ICA assumes that each random variable can be decomposed into a sum of independent random variables. The mixture X_j is a weighted linear combination of K independent components $S_1, S_2, \ldots, S_K$:

$$X_j = a_{j1}S_1 + a_{j2}S_2 + \ldots + a_{jK}S_K. \tag{2.5}$$

Let $\mathbf{X} = (X_1, \ldots, X_J)^T$ be the random vector of mixtures and $\mathbf{S} = (S_1, S_2, \ldots, S_K)^T$ be the random vector of independent components and $\mathbf{A}$ be the mixing matrix with elements a_{jk} with $j \in \{1, 2, \ldots, J\}$ and $k \in \{1, 2, \ldots, K\}$. Then (2.5) can be written in matrix form:

$$\mathbf{X} = \mathbf{AS}. \tag{2.6}$$

All we observed is the mixtures $\mathbf{X}$. Both the hidden variables $\mathbf{S}$ and the mixing matrix $\mathbf{A}$ need to be estimated. Assuming $\mathbf{A}$ is a square matrix which means we have same number of mixing signals and independent components, we can write (2.6) the following way:

$$\mathbf{S} = \mathbf{WX},$$

where $\mathbf{W} = \mathbf{A}^{-1}$ is the inverse of the the mixing matrix. ICA estimates the inverse of mixing matrix by maximizing some measure of independence of $(S_1, S_2, \ldots, S_K)^T$.

2.4.2 ICA for fMRI

There are two types of ICA methods applied to fMRI data, spatial ICA (sICA) and temporal ICA (tICA). sICA assumes the brain areas activated by performance of a certain task should be unrelated to the brain areas whose signals are affected by

artifacts, such as physiological pulsations, subtle head movements and machine noise. At each time point t, sICA decomposes images of the brain $X(t) = (x_1(t), \ldots, x_N(t))$ into K independent components

$$X(t) = a_1(t)S_1 + a_2(t)S_2 + \ldots + a_K(t)S_K, \tag{2.7}$$

where $S_k = (s_{k1}, \ldots, s_{kN})$ is a N-dimension brain image. The coefficients, $a_k(t)$ $t = 1, \ldots, T$, are considered as the activation time series associated with the k-th component. Equation 2.7 implies the change of the observed signal $X(t)$ results from a change in the relative contribution from each component other than from component itself. Activation component is found by computing the correlation between the time series of independent components and a reference function, usually the time course of stimuli and or the expected BOLD signal. The underlying argument is the same: the activated voxels share a similar time course as neural activity.

The activated map is the component whose associated time course has the highest correlation. The voxels with the highest weights on the activated map are considered as activated regions. *McKeown et al.* (1998) first applied sICA to fMRI data and argued that there were spatial independence among consistently task-related fMRI activation (the components that were highly correlated with the reference function), transiently task-related fMRI activation (the components that were correlated with the reference function during part of the trial), slowly varying components (regions of ventricular system), head movement (the components that have abrupt changes in their time courses), quasiperiodic components (signals might be caused by aliased cardiac and respiratory rhythms) and noise components. It also argued that maps of the activated voxels for task-related components contained areas of activation resembling those produced by computing the correlation between observed signal and reference function. In addition, ICA method detected other area that have not detected by the

correlation method.

tICA assumes that the time series of each individual voxel can be decomposed into linear combination of independent times series. For each voxel n, tICA models the time series $X_n = (X_n(1), \ldots, X_n(T))$ as a linear mixture of K independent components $(S_1, \ldots, S_K)$

$$X_n = a_{n1}S_1 + a_{n2}S_2 + \ldots + a_{nK}S_K,$$

where $S_k = (s_k(1), \ldots, s_k(T))$ is a T-dimension time series. The coefficients, a_{nk}, $n = 1, \ldots, N$ is the brain map associated with the k-th component. The activation component is the one with the highest correlation with the reference function. Then activation area can be found by inference about the brain map associated with the activation component. *Biswal and Ulmer* (1999) used tICA to decompose the observed signals into different identifiable individual sources, such as task-related components, cardiac and respiratory pulsations.

Assuming spatial or temporal independence of fMRI data yields two different interpretation of the ICA method. sICA has dominated the application of fMRI. One possible explanation is that standard ICA algorithm needs whitening the data first which projects the mixed signals onto a much smaller K-dimension space. Since in fMRI data set, the spatial dimension (about 100000 voxels) is much larger than temporal dimension (usually 200-300 scans), the preprocessing step for tICA loses too much information about the original data. *Calhoun et al.* (2001) examined these two different approaches. Results showed that sICA and tICA tended to have similar results given components were independent in both space and time and diverged if the components were highly correlated in space or time. It was shown that if there was one single experimental design, both sICA and tICA can separate the BOLD signal from other sources (*Petersen et al.*, 2000). So whether applying sICA or tICA

should depend on the question whether the expected BOLD signals or hypothesized activated areas are heavily dependent.

Several questions need to be addressed before applying ICA. First, the number of independent components we want to extract has to specify before. The results depend heavily on the choice of K and there is no natural ordering of independent components. The standard algorithm sets the number of independent components the same as the number of observed signals. Second, the interpretation of other independent components is not clear. Third, there are several algorithms proposed to find independent components based on different contrast functions. Applying different algorithm might lead to different activation areas. Several popular algorithms are explained briefly in the following sections.

2.4.3 Identifiability Issues

Comon (1994) addressed the identifiability issue of ICA. First, the number of observed mixture signals must be at least as large as the number of independent components. To identify $\mathbf{A}$ and $\mathbf{S}$, we have to put a further constraint that $\text{var}(S_k) = 1$. Since any orthogonal transformation of independent Gaussian variables is also independent, another fundamental assumption in ICA model is that independent components should be non-Gaussian. In order to uniquely determine the independent components, we need the following conditions:

- S_k for $k = 1, 2, \ldots, K$ are non-Gaussian, with possible exception of at most one component.
- $\text{var}(S_k) = 1$ for $k = 1, 2, \ldots, K$.
- The number of mixing signals should be no less than the number of independent components, $J \geq K$.

With these constraints, the mixing matrix $\mathbf{A}$ and the hidden variables $\mathbf{S}$ can be identified up to a permutation matrix (*Hyvärinen and Oja*, 2000).

2.4.4 Measures of Independence

All ICA algorithms are based on the optimization of some measures of independence of $\mathbf{S}$. Popular algorithms used in fMRI data analysis are Infomax (*Bell and Sejnowski*, 1995), JADE (*Cardoso and Souloumiac*, 1993) and FastICA (*Hyvärinen and Oja*, 2000). The performance of different algorithms depends on how well the data's high order structure matches the assumptions of the algorithm. Infomax algorithm works well for sICA (*McKeown et al.*, 1998). However, when the Infomax algorithm looked for temporally independent waveforms, it was less efficient because the boxcar design of the experiment doesn't match its implicit assumption about the underlying distribution. (*McKeown et al.*, 2003).

Maximum Likelihood Approach

One possible way to estimate both independent sources $\mathbf{S}$ and mixing matrix $\mathbf{A}$ is to take a maximum likelihood approach *Pham and Garat* (1997). Under model 2.5, the likelihood of the observed signal can be written as

$$\begin{aligned}\mathcal{L}(\mathbf{A},\mathbf{x}) &= \sum_{n=1}^{N}\sum_{k=1}^{K}\log p_k(\mathbf{w}_k\mathbf{x}_n) + N\log|\det(\mathbf{W})| \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K}\log p_k(e_k^T\mathbf{A}^{-1}\mathbf{x}_n) + N\log|\det(\mathbf{A}^{-1})|.\end{aligned}$$

The likelihood estimate of $\mathbf{A}$ is the value that maximizes $\mathcal{L}(\mathbf{A},\mathbf{x})$.

Information Maximization

Bell and Sejnowski (1995) took an information-maximization (Infomax) approach. This approach is based on maximizing the entropy of a non-linear function of the inde-

pendent sources, Let $\mathbf{H}$ be the entropy function, a direct computation of $\mathbf{H}(S_1, S_2 \ldots, S_K)$ gives

$$\begin{aligned}
\mathbf{H}(\mathbf{S}) &= -\int \mathbf{P}(\mathbf{S}) \log \mathbf{P}(\mathbf{S}) d\mathbf{S} \\
&= -\int \mathbf{P}(\mathbf{X}) \log \frac{\mathbf{P}(\mathbf{X})}{|\det \mathbf{W}|} d\mathbf{X} \\
&= \mathbf{H}(\mathbf{X}) + \log |\det \mathbf{W}|.
\end{aligned}$$

So without any regulation, $\mathbf{H}(\mathbf{S})$ diverges to infinity for an arbitrary large $\mathbf{W}$. Thus, Infomax approach considers entropy of some contrast function $\mathbf{g}$, which is usually an increasing function mapping from $\mathbb{R}$ to $[0, 1]$. The algorithm finds the estimates of $\mathbf{S}$ that maximize $\mathbf{H}(\mathbf{g}(S_1), \mathbf{g}(S_2), \ldots, \mathbf{g}(S_K))$.

Let random variable V_i be a random variable whose cumulative distribution function is $\mathbf{g}$. Let $\mathbf{V} = (V_1, V_2, \ldots, V_K)$ and $\mathbf{U} \sim \mathbf{Unif}[0, 1]^K$. Then the entropy of the contrast function of hidden components is:

$$\begin{aligned}
\mathbf{H}(\mathbf{g}(\mathbf{S})) &= -\mathbf{E}_{\mathbf{g}(\mathbf{S})} \log \mathbf{P}(\mathbf{g}(\mathbf{S})) \qquad (2.8) \\
&= \mathbf{KL}(\mathbf{g}(\mathbf{S})||\mathbf{U}) \\
&= \mathbf{KL}(\mathbf{S}||\mathbf{g}^{-1}(\mathbf{U})) \\
&= \mathbf{KL}(\mathbf{S}||\mathbf{V}).
\end{aligned}$$

This shows the Infomax approach is equivalent to the minimization of the Kullback-Leibler (KL) distance between the independent resources $\mathbf{S}$ and the distribution associated with g. A popular choice of contrast function $\mathbf{g}$ is logistic function, $\mathbf{g}(s) = (1 + e^{-s})^{-1}$ (*Bell and Sejnowski*, 1995).

Cardoso (1997) shows if the contrast function $\mathbf{g}$ is chosen as the cumulative distribution function of $\mathbf{S}$, infomax is equivalent to maximum likelihood estimation. The maximum likelihood approach can be written as a KL distance between two distri-

butions:

$$\begin{aligned}
\arg\max \mathcal{L}(\mathbf{A}, \mathbf{x}) &= \frac{1}{N} \arg\max_{\mathbf{A}} \sum_{n=1}^{N} \log \mathbf{P}(\mathbf{X}_n|\mathbf{A}) \\
&= \arg\max_{\mathbf{A}} \int \mathbf{P}^*_{\mathbf{X}} \log \mathbf{P}(\mathbf{X}|\mathbf{A}) d\mathbf{X} \\
&= \arg\max_{\mathbf{A}} -\mathbf{KL}(\mathbf{P}^*_{\mathbf{X}}||\mathbf{P}(\mathbf{X}|\mathbf{A})) - \mathbf{H}(\mathbf{P}^*_{\mathbf{X}}) \\
&= \arg\min_{\mathbf{A}} \mathbf{KL}(\mathbf{P}^*(\mathbf{S}|\mathbf{A})||\mathbf{P}(\mathbf{S})),
\end{aligned}$$

where $\mathbf{P}^*_{\mathbf{X}}$ is the empirical distribution of $\mathbf{X}$ and $\mathbf{P}^*(\mathbf{S}|\mathbf{A})$ is the empirical distribution of $\mathbf{S}$ given mixing matrix $\mathbf{A}$. So, if the contrast function $\mathbf{g}$ in (2.8) is chosen to be the cumulative distribution function of $\mathbf{S}$, then the maximum likelihood method is the same as Infomax approach. In the infomax approach, any contrast function $\mathbf{g}$ mapping from $\mathbb{R}$ to $[0, 1]$ is chosen as the cumulative distribution function of the independent sources which need to be estimated in the maximum likelihood method.

Mutual Information and Kullback-Leibler divergence

Comon (1994) used mutual information as a measure of dependence. The mutual information $\mathbf{I}$ of K random variables $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_K)$ is defined by the following equation:

$$\mathbf{I}(Y_1, Y_2, \ldots, Y_K) = \sum_{k=1}^{K} \mathbf{H}(Y_k) - \mathbf{H}(\mathbf{Y}).$$

Mutual information can be interpreted as the a measure of the information that $Y_1, Y_2, \ldots, Y_T$ share. It is always non-negative and is zero if and only if $(Y_1, Y_2, \ldots, Y_T)$ are independent. So the ICA estimate of the problem 2.6 using the mutual information is:

$$\arg\min_{\mathbf{A}} \mathbf{I}(e_1^T \mathbf{A}^{-1}\mathbf{X}, e_2^T \mathbf{A}^{-1}\mathbf{X}, \ldots, e_K^T \mathbf{A}^{-1}\mathbf{X}).$$

Since the entropy depends on the unknown distribution of $\mathbf{S}$, the maximization of the mutual information needs the approximation of the density function. *Comon*

(1994) used Gram-Charlier expansion which is a polynomial density expansion based on higher-order cumulants. For random variable Y of zero mean and unit variance, the Gram-Charlier expansion is

$$\mathbf{P}(y) \approx \phi(y)(1 + \kappa_3(Y)h_3(y)/6 + \kappa_4(Y)h_4(y)/24 + \ldots),$$

where ϕ is the Gaussian density function. $\kappa_i(Y)$ is the i-th cumulants of the random variable Y and $h_i(y)$ are Hermite polynomials defined recursively:

$$\begin{aligned} h_0(y) &= 1 \\ h_1(y) &= y \\ h_{n+1}(y) &= yh_n(y) - h'_n(y) \end{aligned}$$

Plugging the estimate of the density function, we get the mutual information of $\mathbf{S} = (s_1, s_2, \ldots, s_K)$, under the constraint that $(s_1, s_2, \ldots, s_K)$ are uncorrelated. We have:

$$\mathbf{I}(\mathbf{S}) = C + \frac{1}{48}\sum_{k=1}^{K}(4\kappa_3(s_k)^2 + \kappa_4(s_k)^2 + 7\kappa_4(s_k)^4 - 6\kappa_3(s_k)^2\kappa_4(s_k)), \tag{2.9}$$

where C is a constant and $\kappa_i(s_k)$ is the i-th cumulant of empirical distribution $\mathbf{P}_k^*$ of s_k.

Since the Gram-Charlier expansion is based on Taylor expansion of density function at the point of Gaussian density function, the approximation is valid if the true distribution is not far from Gaussian. Then the estimate achieves the minimum of (2.9).

Kurtosis

One-contrast function method allows the estimation of one independent component at each time. Instead of maximizing some measures of mutual independence, one-contrast function method tries to maximize the measure of non-Gaussianity for each component S_k. In many applications, we are only interested in a few components. So it is not necessary to extract K independent components at the same time (*Hyvärinen*, 1999). And estimation of one component at one time greatly reduces the computation complexity.

From the projection pursuit point of view, the decomposition of J signals into a weighted sum of K components is to project high dimension data onto a lower space. It has been argued that Gaussian distribution is the least interesting structure in terms of the information it carries. So the projection should be in the least Gaussian direction. One-contrast function method uses the same idea trying to find the least Gaussian projection at each time.

The classical measure of non-Gaussianity is kurtosis. The kurtosis of random variable Y is the fourth cumulant:

$$\text{Kurt}(Y) = \mathbf{E}(Y^4) - 3(\mathbf{E}(Y^2))^2.$$

If Y has unit variance, then $\text{Kurt}(Y) = \mathbf{E}(Y^4) - 3$ is a normalized version of the fourth moment. For standard Gaussian variable z, $\text{Kurt}(z) = 0$. For most non-Gaussian variables kurtosis is nonzero. Kurtosis is widely used as a measure of non-Gaussianity. The main reason is its linearity property which makes both theoretical and computational analyses easier. For independent variables Y_1 and Y_2 with zeros means and unit variances,

$$\text{Kurt}(c_1Y_1 + c_2Y_2) = c_1^4\text{Kurt}(Y_1) + c_2^4\text{Kurt}(Y_2).$$

Let $\mathbf{w}$ be a row vector in the inverse mixing matrix $\mathbf{W}$. Then $\mathbf{wX}$ is the estimate of one component. The kurtosis approach tries to maximize the kurtosis of the estimated component,

$$\begin{aligned}\max_{\mathbf{w}} |\text{Kurt}(\mathbf{wX})| &= \max_{\mathbf{w}} |\text{Kurt}(\mathbf{wAS})| \\ &= \max_{\mathbf{c}} |\text{Kurt}(\mathbf{cS})| \\ &= \max_{\mathbf{c}} |\sum_{k=1}^{K} c_k^4 \text{Kurt}(S_k)|,\end{aligned}$$

where $\mathbf{c} = \mathbf{wA}$. Since we assume that $\text{var}(S_k) = 1$ for $k = 1, 2, \ldots, K$, we get $\sum_{k=1}^{K} c_k^2 \text{var}(S_k) = \sum_{k=1}^{K} c_k^2 = 1$. So the optimization problem becomes

$$\max_{\mathbf{c}} |\sum_{k=1}^{K} c_k^4 \text{Kurt}(S_k)| \quad \text{with the constraint} \quad \sum_{k=1}^{K} c_k^2 = 1, \tag{2.10}$$

If we assume there is at least one component whose kurtosis is negative and at least one whose kurtosis is positive, the maximum in (2.10) is achieved at $\mathbf{c} = \pm e_j^T$, where e_j is a column vector with 1 on the j-th row and 0 elsewhere (*Delfosse and Loubaton*, 1995). Then $\mathbf{wX} = \pm e_j^T \mathbf{A}^{-1}\mathbf{X} = \pm e_j^T \mathbf{S}$. This means maximizing the contrast function gives us the independent component S_j up to a sign difference. Since the measure of non-Gaussianity is based on the fourth cumulant, the kurtosis approach is very sensitive to outliers.

Negentropy

Negentropy is a natural choice to assess the distance between Gaussian distribution and any other distribution. Let $\mathbf{J}$ be the negentropy of any random vector $\mathbf{S}$, negentory of $\mathbf{J}$ is defined as

$$\mathbf{J}(\mathbf{S}) = \mathbf{H}(\mathbf{S}_{gauss}) - \mathbf{H}(\mathbf{S}),$$

where $\mathbf{S}_{gauss}$ is a Gaussian random vector which has the same mean and covariance matrix as $\mathbf{S}$. Negentropy is non-negative and achieves 0 if and only if $\mathbf{S}$ is Gaussian. It is invariant to any linear transformation.

There is a natural link between negentropy and independence through mutual information. The Mutual information can be expressed in terms of negentropy,

$$\mathbf{I}(S_1, S_2, \ldots, S_K) = \mathbf{J}(\mathbf{S}) - \sum_{k=1}^{K} \mathbf{J}(S_k) + \frac{1}{2} \log \frac{\prod_{k=1}^{K} \Sigma_{kk}}{\det(\Sigma)|}, \tag{2.11}$$

where Σ is the covariance matrix of $\mathbf{S}$ and Σ_{kk} is its k-th diagonal element. If $(S_1, S_2, \ldots, S_K)$ are uncorrelated then (2.11) becomes

$$\mathbf{I}(S_1, S_2, \ldots, S_K) = \mathbf{J}(\mathbf{S}) - \sum_{k=1}^{K} \mathbf{J}(S_k).$$

In the ICA model (2.6), the negentropy of $\mathbf{S}$ is the same as the negentropy of $\mathbf{X}$ which does not depend on $\mathbf{W}$. Maximizing independence is the same as minimizing mutual information and also the same as maximizing the sum of negentropy. Assuming $S_1, S_2, \ldots, S_K$ are uncorrelated,

$$\begin{aligned} \arg\min_{\mathbf{A}} \mathbf{I}(\mathbf{S}) &= \arg \max_{\mathbf{w}_1, \ldots, \mathbf{w}_K} \sum_{k=1}^{K} \mathbf{J}(S_k) \\ &= \arg \max_{\mathbf{w}_1, \ldots, \mathbf{w}_K} \sum_{k=1}^{K} \mathbf{J}(\mathbf{w}_k \mathbf{X}), \end{aligned}$$

with the constraint that $(\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K)$ are linearly independent where $\mathbf{w}_k$ is the k-th row in the inverse mixing matrix $\mathbf{W}$. Negentropy of S depends on the unknown distribution of S. Different approximations were proposed based on different assumptions of the underlying distribution. *Jones and Sibson* (1987) used Gram-Charlier expansion as an approximation of density function to compute negentropy.

Given S is of zero mean and unit variance, the negentropy has the following form:

$$\mathbf{J}(S) \approx \frac{1}{12}\kappa_3(S)^2 + \frac{1}{48}\kappa_4(S)^2,$$

where κ_3 and κ_4 are the third and fourth cumulants of S. This approximation is also a polynomial function of cumulants. It has been argued that these cumulant-based methods often provide a poor approximation of entropy since higher order cumulants are quite sensitive to outliers.

Hyvärinen (1998) proposed a different approximation of negentropy. Given S has zero mean and unit variance.

$$\mathbf{P}(S) \approx \phi(S)(1 + \sum_{1}^{n} c_i \mathbf{G}_i(S)),$$

where $\mathbf{P}(S)$ is the density function of S, ϕ is the standard Gaussian density function and $\mathbf{G}_i$ are some regulation functions which satisfy

$$\begin{aligned} \int \mathbf{P}(S)\mathbf{G}_i(S)dS &= c_i \quad \text{for} \quad i = 1, 2, \ldots, n \\ \int \phi(S)\mathbf{G}_i(S)\mathbf{G}_j(S)dS &= \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{if} \quad i \neq j \end{cases}. \end{aligned}$$

Then using Taylor approximation to the logarithmic function $(1 + \mathbf{P}(S))\log(1 + \mathbf{P}(S)) = \mathbf{P}(S) + \mathbf{P}(S)^2/2$, the approximation to negentropy is:

$$\mathbf{J}(S) \approx \sum_{i=1}^{n} k_i[\mathbf{E}(\mathbf{G}_i(S)) - \mathbf{E}(\mathbf{G}_i(z))]^2, \tag{2.12}$$

where k_i are constants and z follows a standard Gaussian distribution.

Theoretically, $\mathbf{G}_i$ can be any orthogonal function with respect to Gaussian distribution. But in practice, the expectation of $\mathbf{G}_i(S)$ should be easy to compute. And

in order to get more robust estimate than cumulant approach, $\mathbf{G}_i(S)$ must not grow faster than quadratically. *Hyvärinen and Oja* (2000) considered the simplest case when $n = 1$. Then (2.12) becomes

$$\mathbf{J}(S) \approx [\mathbf{E}(\mathbf{G}(S)) - \mathbf{E}(\mathbf{G}(z))]^2.$$

Two possible choices for $\mathbf{G}$ are given, (*Hyvärinen and Oja*, 2000)

$$\mathbf{G}_1(S) = \frac{1}{a} \log \cosh aS \quad \mathbf{G}_2(S) = -\exp(-S^2/2),$$

where a is some suitable constant, usually $1 \leq a \leq 2$.

2.5 Gaussian Process

At the modeling level, GLM assumes spatio-temporal independence in fMRI data which is generally not a reasonable assumption. Spatial correlation is addressed indirectly by smoothing the data using a Gaussian kernel in the preprocessing step and then applying Gaussian RFT to the map of test statistics. The difference in the assumptions of two-level analysis makes standard model diagnosis not feasible. Models that incorporate the spatio-temporal dependences are desirable. $\mathbf{AR}(1)$ plus white noise model takes the temporal correlation into consideration by specifying the first order correlation for all voxels. However, the order of temporal correlation depends on several factors which can vary across different regions in brain. We propose a model using the Gaussian process to accommodate this variability in temporal correlation.

2.5.1 Gaussian Process for fMRI

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution. A Gaussian process can be completely specified by its mean function $m(t)$ and the covariance function $k(t, t')$. Let $f(t)$ be a Gaussian process with mean function $m(t)$ and covariance function $k(t, t')$,

$$\begin{aligned} m(t) &= \mathbf{E}f(t) \\ k(t, t') &= \mathbf{E}(f(t) - m(t))(f(t') - m(t')). \end{aligned}$$

Then Gaussian process $f(t)$ denotes as

$$f(t) \sim \mathcal{GP}(m(t), k(t, t'))$$

Neal (1998) used the Gaussian process model for both regression and classification showing that it is a very flexible method to define prior distributions over functions. He argued that the there are several reasons for its popularity. First, a variety choices of covariance functions can give functions in different degrees of smoothness. Second Gaussian process is suited for modeling of large number of correlated variables. Because of the explicit form of conditional gaussian distribution, the estimation is much easier than other distributions. Third, it is easy to incorporate the prior information into the Gaussian process. These advantages make Gaussian process a useful tool for the fMRI analysis. So In our model, we proposed a different way to decompose the signal into a long drift signal, an activation signal, a temporal correlated signal and an independent noise. The variation in the signal were divided into the Gaussian process and the pure noise.

Let $X(t)$ be the time series for a specific voxel, the signal can be decomposed into a linear combination of mean function $m(t)$, a zero mean Gaussian process $G(t)$ and

a white noise term $\epsilon(t)$

$$X(t) = \beta_0 + \beta_1 I(t) + \beta_2 g(t) + G(t) + \epsilon(t). \tag{2.13}$$

β_0 is the parameter of the scale of time series. The intensity of fMRI image depends on the type of tissue it measures, ranging from 0 (area outside of brain) to about 1600 (cerebrospinal fluid area). $I(t)$ is a centered linear function which characterizes the linear trend usually observed in fMRI signals. $g(t)$ is the expected BOLD signal evoked by the experimental stimulation, modeled as a convolution of stimulus function and the canonical HRF. $\epsilon(t)$ is the white noise, $\epsilon(t) \sim \mathcal{N}(0, \sigma_\epsilon^2 \mathbb{I})$.

The term $G(t)$ is a zero mean Gaussian process, which characterizes the temporally correlated component in $X(t)$, such as physiological effect and random drift due to instability of scanner. The temporal correlation should decrease as the time lag increases. Further it is reasonable to assume the effect of series correlation only exits in a relative short-range and the order of this correlation varies across the different regions of the brain. Based on the above assumption, we use an exponential covariance matrix with two parameters σ^2 and ϕ to characterize the Gaussian process.

$$G(t) \sim \mathcal{GP}(0, \sigma^2 \exp(\frac{-|t - t'|}{\phi})),$$

where σ^2, the variance of Gaussian process, measures the amount of fluctuation. Parameter ϕ controls the order of time correlation. As ϕ decreases, the temporal dependence goes to 0 at a very fast rate.

The mean function of Gaussian process model (2.13) is similar to the one in GLM. The covariance function of Gaussian process addresses the non sphericity in fMRI data set. It decomposes the noise term in equation (2.1) into two zero mean Gaussian components. One is a temporally independent component which is just random noise. The other is a temporally dependent signal which can reveal some

information of functional regions of the brain. Since there is a parameter ϕ in the covariance matrix, the estimation needs some iterative fitting techniques or sampling methods. A simulation study is done applying both Metropolis-in-Gibbs sampler and Expectation-Maximization (EM) method to examine whether Gaussian process can be separated from noise term.

2.5.2 Simulation Study

Time series with a length of 240 is generated from (2.13) with six parameters chosen to match the real data. The scale parameter, $\beta_0 = 600$, reflects the intensity of signals of grey matter in the brain. The linear trend parameter, β_1, is set to 0.01. The activation parameter, $\beta_2 = 20$, characterizes strong activation. Total variance is set to be $\sigma^2 + \sigma_\epsilon^2 = 100$. The simulation study shows that the performance of evaluation of the model depends mainly on the values of two quantities, temporal correlation parameter, ϕ, and ratio of two variance, $\kappa = \frac{\sigma_\epsilon^2}{\sigma^2}$. Different combination is investigated. ϕ is set at three different levels, little correlation $\phi = 0.1$, modest correlation $\phi = 1$ and high correlation $\phi = 4.5$. And κ is investigated at three levels, noise variance dominating, $\kappa = 4$, equal variance $\kappa = 1$ and Gaussian process variance dominating, $\kappa = 0.25$.

Metropolis-in-Gibbs sampler

For Bayesian sampler method, non-informative priors were put for all six parameters. Three parameters β_0, β_1 and β_2 in the mean function are sampled by Gibbs algorithm and three parameters ϕ, σ^2 and σ_ϵ^2 in the covariance function are sampled by Metroplis-Hasting algorithm. Figure 2.3 shows the posterior distribution for three mean parameters. From left to right the posterior distribution of β_0, β_1, β_2

For $\phi = 1$, $\sigma^2 = 50$, $\sigma_\epsilon^2 = 50$, the posterior distributions of parameters, β_0, β_1 and β_2, peak around the true value. Figure 2.4 shows the posterior distribution for three

covariance parameters. From left to right the posterior distribution of σ^2, σ_ϵ^2, ϕ. The posterior distribution of parameters show reasonable estimation. From left to right the posterior distribution of σ^2, σ_ϵ^2, ϕ

But for small ϕ ($\phi = 0.1$) the loglikelihood is a function of $(\sigma^2 + \sigma_\epsilon^2)$. The algorithm can not differentiate Gaussian process and white noise. Figure 2.5 shows the posterior distribution of $\sigma^2 = 80$ and $\sigma_\epsilon^2 = 20$ when $\phi = 0.1$. From left to right the posterior distribution of σ^2, σ_ϵ^2, ϕ

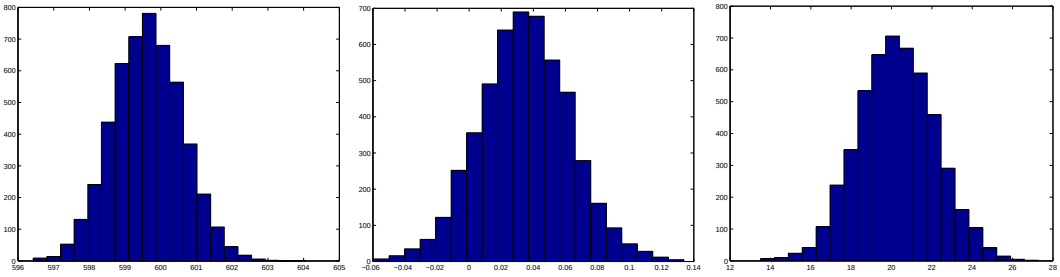

Figure 2.3: The posterior distribution of β_0, β_1, β_2

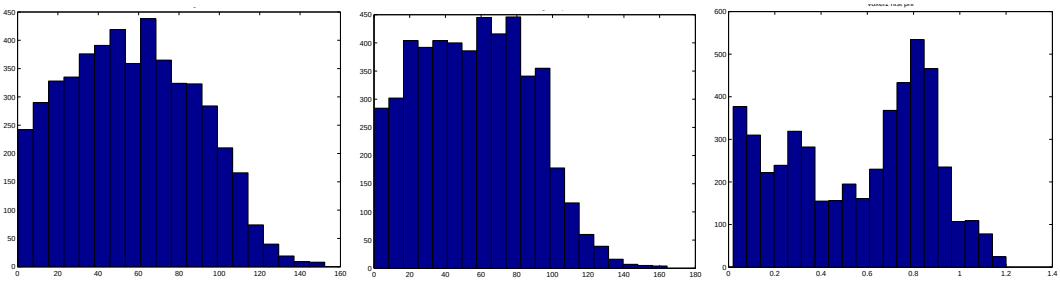

Figure 2.4: The posterior distribution of σ^2, σ_ϵ^2, ϕ

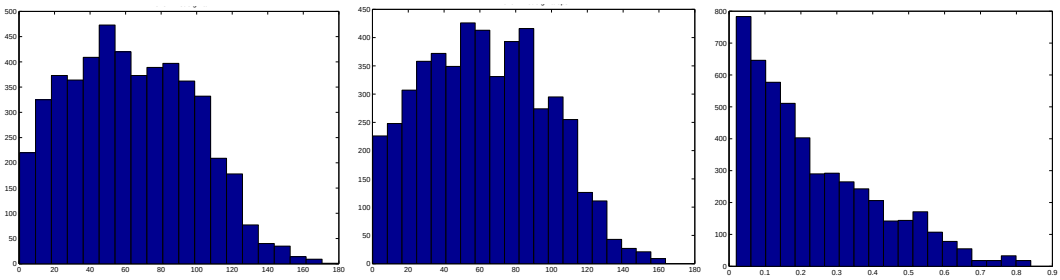

Figure 2.5: The posterior distribution of σ^2, σ_ϵ^2, ϕ

σ^2	σ_ϵ^2	ϕ	$\widehat{\sigma^2}$	$\widehat{\sigma_\epsilon^2}$	$\widehat{\phi}$	$\widehat{\beta_0}$	$\widehat{\beta_1}$	$\widehat{\beta_2}$
20	80	0.1	47.46	51.55	0.45	600.97	0.0267	20.28
80	20	0.1	46.78	59.86	0.03	600.84	0.0102	20.24
20	80	1.0	39.47	49.02	0.55	599.83	0.0125	19.53
80	20	1.0	87.67	6.98	0.97	599.70	0.0229	23.30
20	80	4.5	30.75	83.62	3.24	598.69	0.0173	21.58
80	20	4.5	69.13	25.34	5.00	604.86	0.0287	18.13

Table 2.1: Estimation of parameters for different σ_ϵ^2 and σ^2

Expectation-Maximization Algorithm

Fixing the parameters of mean function, $\beta_0 = 600$, $\beta_1 = 0.01$ and $\beta_2 = 20$ and also the total variance $\sigma^2 + \sigma_\epsilon^2 = 100$, EM algorithm is applied to time series generated from different values of $\kappa = \sigma_\epsilon^2/\sigma^2$ and ϕ. where σ^2, σ_ϵ^2 and ϕ are true values which generate the data and $\widehat{\sigma^2}$,$\widehat{\sigma_\epsilon^2}$ and $\widehat{\phi}$ are the estimated values by EM algorithm. The results show that the EM algorithm can get good estimate of mean function in all the cases. When ϕ is small which means there is little correlation in Gaussian process, $G(t)$ behaves like white noise term $\epsilon(t)$. Estimation technique based on distribution can not separate these two terms well. As in the case $\phi = 0.1$, the algorithm tends to decompose the variance equally between σ^2 and σ_ϵ^2. In the case that κ is large, which means the variance of noise dominates, the estimation of ϕ becomes worse. This could due to the reason if the strength of the Gaussian process is small, the data can be corrupted by the noise term which leads to inaccurate estimation.

2.6 Real Data Analysis

Before statistical analysis, there are several preprocessing steps that try to reduce noise from different sources. The major steps involved in fMRI preprocessing are slice timing correction, realignment, coregistration of structural and functional images, normalization and smoothing. It is typically assumed in statistical analysis that all the voxels are collect at the same time. But the slices of brain are sampled sequentially.

Therefore time series of voxels in different slices are shifted relative to each other. Slice timing correction usually uses interpolation method to shift the time series of each voxel. The largest source of noise in any fMRI study is from head movement of subject. When movement occurs, the signal from a specific voxel will be contaminated by signal from neighbors. Motion correction is a rigid body transformation (shifting and rotation) between one image and a target image (usually the first image or the mean image). Usually this is done by minimizing some cost function that measures the similarity between these two images. fMRI images usually sacrifice spatial resolution to achieve a better temporal resolution. Another preprocessing step, coregistration is to map the fMRI image to a structural image of the same subject to make inference about activation. This is typically performed using rigid body transformation or affine transformation (shifting, rotation and scaling). In multiple subjects analysis, it is important that each voxel lies in the same function region to compare the results from different subjects. Normalization attempts to register each subjects structural image to a template brain image. Usually this is done by a nonlinear transformation in two steps. The first step is to estimate a continuous mapping between the points in an input image with those in the target image. Next the mapping is used to resample the input image so that it is warped into the target image. The last step is to spatially smooth fMRI data using a Gaussian kernel.

2.6.1 Experiment Paradigm

One data set was used to assess the performance of different methods. The data was from an auditory stimulation experiment. The experimental paradigm consisted of 16 blocks alternated between rest and auditory stimulation, starting from rest. During auditory stimulation block, subject was listened to bi-syllabic words presented binaurally at a rate of 60 per minute. 6 scans were acquired for each block with one scan taking 7 seconds. A total 96 acquisition were made from a single subjects with

each acquisition with dimension $64 \times 64 \times 64$. The voxel size is $3 \times 3 \times 3$ mm^3. The preprocessing steps involve realignment, coregistration, normalization and smoothing. During preprocessing, interpolation and resampling procedures were applied to each scan which leads to new voxel size $2 \times 2 \times 2$ mm^3. The dimension of preprocessed scan is $79 \times 95 \times 68$.

2.6.2 Activation Analysis

Both GLM and ICA were applied to one slice of the auditory data set. GLM used the canonical HRF with **AR**(1). Spatial ICA was used to detect the activation area for ICA approach. Figure 2.6 shows the activation area map. The left figure is the activation area detected by GLM with Gaussian random field correction at 5% significant level. The right figure is activation area detected by sICA. After applying sICA 95% quantile of activated map was set as the threshold to define the activation area. These figures show a consistency between GLM and sICA. Figure 2.7 shows the area that are mostly related to the experiment. The left figure is the area where the GLM-estimated parameters exceed the 95% quantile of the parameter map. The right figure is the area where Gaussian-process-estimated parameters exceed the 95% quantile of parameter map. The area detected through estimation of parameters matches the activation area. Since the Gaussian process models a higher order temporal correlation, the area detected by Gaussian process is smoother than GLM.

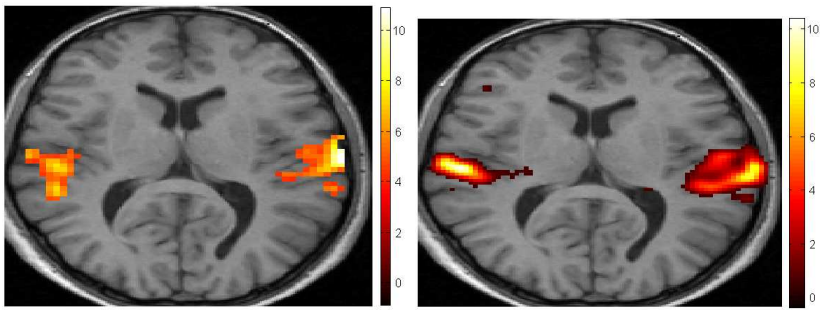

Figure 2.6: Activation maps of GLM and sICA

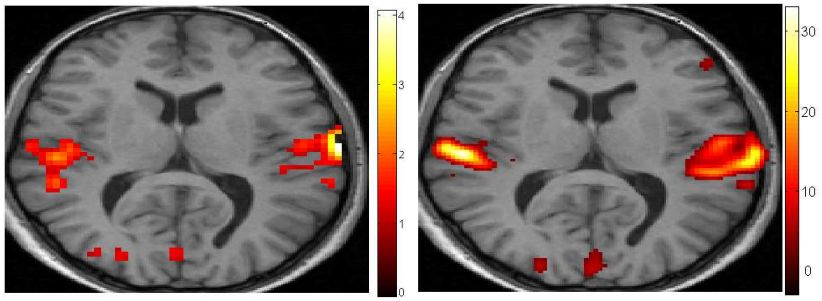

Figure 2.7: Map of β_1 parameter

2.6.3 Gaussian Process Results

One activated voxel and one inactivated voxel are chosen from this slice. Gaussian process is applied to both time series separately. There are three components in the mean function which are estimated by β_0, β_1 and β_2. There are two random components, Gaussian process ($G(t)$) and white noise ($\epsilon(t)$). The estimated Gaussian process and white noise shown in the Figure 2.8 are the conditional expectation given all the parameters and observed signal of the activated voxels and Figure 2.9 are the conditional expectation of $\mathbf{E}(G|X, \beta_0, \beta_1, \beta_2, \sigma^2, \sigma_\epsilon^2, \phi)$ and $\mathbf{E}(\epsilon|X, \beta_0, \beta_1, \beta_2, \sigma^2\sigma_\epsilon^2, \phi)$. Figure 2.8 and Figure 2.9 illustrate the ratio between Gaussian process and white noise. In both cases, the activation parameter β_2 is well estimated.

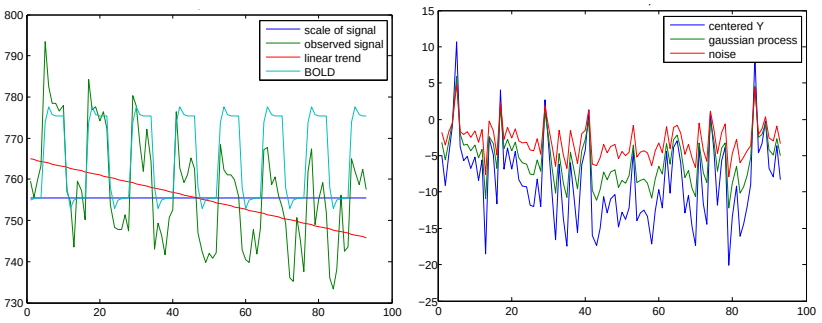

Figure 2.8: Fitted components of an activated voxel. Left figure: estimated mean function. Right figure: estimated random components

The figure illustrates the ratio between Gaussian process and white noise. In both cases, the activation parameter β_2 is well estimated.

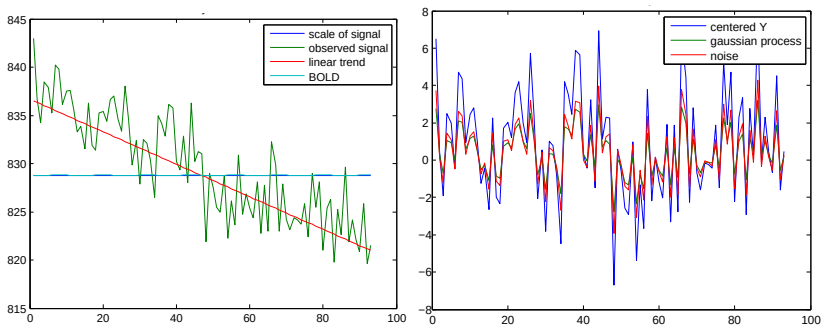

Figure 2.9: Fitted components of an inactivated voxel. Left figure: estimated mean functions. Right figure: estimated random components

2.6.4 Parameter Maps

EM algorithm is applied to the whole slice to assess the performance of the model. The resulting maps of six parameters are imposed on a structure image.

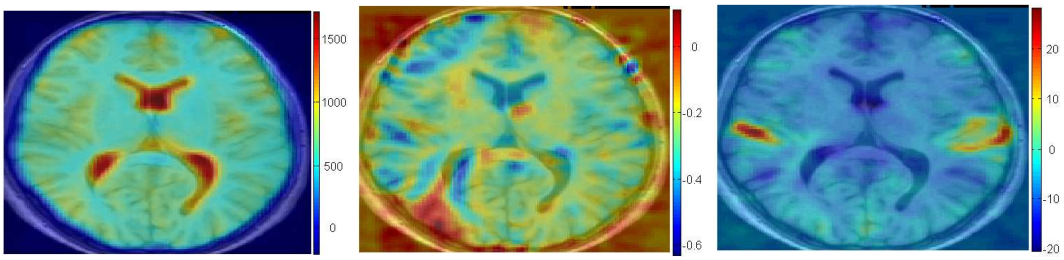

Figure 2.10: Maps of β_0, β_1, β_2, imposed on structure image

The scale map (β_0) (left figure in Figure 2.10) matches the structure image well which captures the scale of different tissue. The linear trend map (β_1) shows there exits a linear trend across the whole slice. It tends to be positive outside the brain and negative inside the brain. The standard way of dealing with this linear trend is to subtract the mean of the whole scan at each time point. This can introduce other confounding factor in the following analysis since this linear trend is not uniform across the brain. The activation map (β_2) shows a smooth image of activation area which is similar to the results of GLM.

The temporal correlation map ϕ (left figure in 2.11) shows that most regions have temporal correlation at order below 4. (Most ϕ are smaller than 2 which give

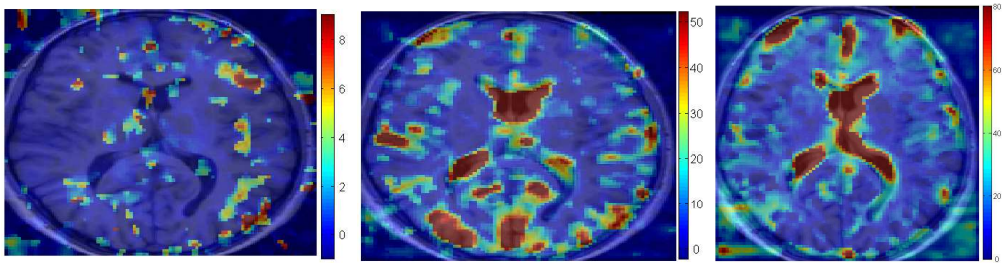

Figure 2.11: Maps of ϕ, σ^2 and σ_ϵ^2, imposed on structure image

correlation $\exp(-4/2) = 0.1352$ at order 4.) There are several regions showing high temporal correlation which need further interpretation.

The model divides total variance into Gaussian process variance and white noise variance. The white noise variance is relatively uniform across the whole slice. The Gaussian process variance tends to be higher for voxels inside the brain than the ones outside the brain. This suggests that the main source for temporal correlation is related to cerebral blood flow.

2.7 Discussion

This chapter proposes a Gaussian process method to model the fMRI signals. The method decomposes the observed signals into the stimulus-related components, a Gaussian process and a white noise. The temporal correlation is modeled by the nonspherical structure of the Gaussian process.

The parameters can be estimated either through the EM algorithm or the Metropolis-Hasting algorithm. The simulation study shows that the model is well-defined and the mean functions can be estimated well in all the scenarios in the simulation study. However when the temporal correlation in the Gaussian process is weak, the Gaussian process can not be separated from the noise. Then the estimates of parameters of variance become unstable.

In the real data analysis, the estimates of the mean function are comparable to the results of the GLM. The scale parameter matches the brain tissue type and

the activation parameter detects the same regions as GLM. Moreover, the regions identified by Gaussian process model has smoother boundaries than GLM.

The estimates of the variance parameters show that the correlation levels vary across the whole brain. Using the autocorrelation model with the same order for the every voxel may not capture the temporal correlation well. The voxels with high correlation are clustered together in the brain. Those regions need further interpretation.

CHAPTER III

Structural MRI Analysis

3.1 Introduction

Structural MRI provides physicians and researchers a noninvasive method to produce high-resolution images of the brain's anatomical structure. The pathological changes associated with neurological and psychiatric diseases may cause loss of brain tissue or atrophy in the brain. Structural MRI offers a way of visualization of these brain changes in vivo by measuring the tissue density at a very fine grid. Different methods have been proposed to analyze the structural MRI and a number of studies have already demonstrated that MRI scans can provide biological plausible results in various diseases (*Kopelman et al.*, 2001), (*Bottino et al.*, 2002) and (*Shenton et al.*, 1991).

In the following sections, we presented two popular methods, voxel-based method (*Wright et al.*, 1995) and support vector machine (SVM) (*Klöppel et al.*, 2008) in more details for structural MRI analysis. We discovered each method's advantages and limitations. Then we proposed a new method for structural MRI based on multiple kernel SVM (*Sonnenburg et al.*, 2006). Theoretically multiple kernel SVM is an extension from single kernel SVM but the this extension gives more flexibility to the method leading to a better classifier in many cases. We then study the new method on both simulated data and real data. The performance of the method is discussed under different scenarios.

3.2 Voxel-based Method

The traditional technique of identifying structural changes in the brain is a volumetric measurement method, involving manually drawing regions of interests (ROI) and visually assessing any morphological changes in those regions (*Chan et al.*, 2001), (*Keller and Roberts*, 2009). However, as MRI scans become a standard procedure for both clinical diagnosis and brain research, automated tools are desired to save time

and energy from time-consuming manual measurements and subjective assessment. Voxel-based morphometry (VBM) is such a technique proposed by Wright in 1995 (*Wright et al.*, 1995). This method first maps all the scans to a brain template and then constructs a statistical test for every voxel to identify the regional differences between the two groups. It is the counterpart of the GLM in the functional MRI analysis and quite successful in distinguishing neurodegenerative diseases (*Whitwell and Jack*, 2005).

Registeration VBM is a univariate method, comparing the values of one voxel across multiple scans at one time. In order for the statistical tests to be valid, all brain scans have to be registered to the same space. Then one voxel from one scan will mean the same voxel of other scans. This step is called spatial registration which involves a rigid body transformation (*Friston et al.*, 1995a) and a non-linear warping (*Ashburner and Friston*, 1999). The rigid body transformation optimizes an affine transformation that maps the individual MRI scan to a template. This corrects for the head movement of different subjects. The non-linear warping involves mapping images of individuals into the same template through a set of basis functions. This reduces the variability of different shapes of the brains. After registration, the scans are aligned to the same template and a location in one scan corresponds to the same location in another scan. However, registration also reduces the disease-related morphometric differences between two groups which are the signals that we want to detect in VBM.

Segmentation Some neuro-related diseases will cause the shrinkage in the volume of gray matter and the expansion of the white matter in local regions of the brain. Segmenting the brain into different tissues will facilitate the detection of the affected regions and minimize the partial volume effects. One popular procedure before statistical analysis is segmenting the brain into gray matter, white matter and cerebrospinal

fluid (*Fischl et al.*, 2002). One way is to use K mixtures of Gaussian to model the voxel density while building the voxel location information into the prior information (*Ashburner and Friston*, 2005). This method is explained below.

Let μ_k and σ_k^2 be the mean and the variance of the k-th Gaussian of the whole brain. Let c_i be the class label of voxel i, $c_i \in \{1, \ldots, K\}$. y_i is the value of the i-th voxel. Then the conditional probability of voxel i given that the voxel belongs to the k-th Gaussian is

$$P(y_i|c_i = k, \mu_k, \sigma_k^2) = \frac{1}{(2\pi\sigma_k^2)^{\frac{1}{2}}} \exp\left(-\frac{(y_i - \mu_k)^2}{2\sigma_k^2}\right).$$

Let γ_k be the mixture proportion for the k-th Gaussian, $\sum_{k=1}^{K} \gamma_k = 1$. Rather than assuming a stationary prior probability across the whole brain, the prior takes the voxel location into consideration:

$$P(c_i = k|\gamma_{\mathbf{k}}, \alpha) = \frac{\gamma_k b_{ik}(\alpha)}{\sum_{j=1}^{K} \gamma_j b_{ij}(\alpha)},$$

where b_{ik} is a function incorporating the tissue probability for class k at voxel i. α is the deformation parameters of a set of spatial basis functions. Then the log-likelihood function for a single voxel i can be written as

$$\begin{aligned}
\mathcal{L}(y_i|\mu, \sigma^{\mathbf{2}}, \gamma, \alpha) &= \log\left(\sum_{k=1}^{K} P(y_i, c_i = k|\mu, \sigma^{\mathbf{2}}, \gamma, \alpha)\right) \\
&= \log\left(\sum_{k=1}^{K} P(y_i|c_i = k, \mu_k, \sigma_k^2) P(c_i = k|\gamma_{\mathbf{k}}, \alpha)\right) \\
&= -\log\left(\frac{1}{\sum_{k=1}^{K} \gamma_k b_{ik}(\alpha)} \sum_{k=1}^{K} \frac{\gamma_k b_{ik}(\alpha)}{(2\pi\sigma_k^2)^{\frac{1}{2}}} \exp(-\frac{(y_i - \mu_k)^2}{2\sigma_k^2}\right).
\end{aligned}$$

The parameter estimates of $\mu, \sigma^{\mathbf{2}}, \gamma, \alpha$ are the MLE of the joint log-likelihood of the

all the voxels which maximize the following log-likelihood function $\mathcal{L}$

$$\mathcal{L}(\overrightarrow{y}|\mu, \sigma^{\mathbf{2}}, \gamma, \alpha) = \sum_{i=1}^{V} \mathcal{L}(y_i|\mu, \sigma^{\mathbf{2}}, \gamma, \alpha),$$

where $\overrightarrow{y} = \{y_1, \ldots, y_V\}$ The probability $P(c_i = k)$ serves as an indicator of the portion of the tissue k in the voxel i which can be used as the features in the further analysis. After segmentation step, a gray matter image and a white matter image is produced with the values of the probabilities.

Smoothing After the segmentation step, the gray and white images are smoothed by convolving with a three dimensional Gaussian kernel. The smoothing step helps to reduce the effect of the noise in the original image. It also compensates for the inexact nature of the spatial registration and segmentation in previous step.

Statistical Test After preprocessing step, a statistical test between two group means is applied to every voxel in the image (*Friston et al.*, 1995b). This involves applying a t-test or a F-test, taking any covariates into consideration. The result is a statistical parameter map of the whole brain with a p-value for each voxel. The clusters of voxels with small p-values may be regions that are associated with the disease and need further inspection. Since the statistical parametric map contains the p-values of correlated voxels, multiple test correlation is needed when assessing the significance in any voxel.

Although the voxel-based methods had been widely used to study the morphological changes in the brain, some scholars discussed its limitations, suggesting to use it with great caution (*Mechelli et al.*, 2005). First, the voxel-based method is a univariate method, which means it considers one voxel at a time and predicts the voxel as significant or not only based on the signals at that voxel. So the VBM is more likely to discover the changes that are localized in space, overlooking the differences in brain

networks (*Davatzikos*, 2004). Second, VBM depends on the t-test or the F-test. The validation of these statistical tests rely on the assumption that the residuals have independent Gaussian distributions. The non-normality distribution will attenuate the power of the tests (*Salmond et al.*, 2002). And finally, the pre-processing steps of registration and normalization can bring noise to the data (*Gitelman et al.*, 2001). Mapping a brain containing pathologies changes to a standard template may mask the true differences between the two groups (*Mechelli et al.*, 2005).

3.3 Machine Learning Methods

Based on the limitations of univariate methods, multivariate methods are proposed to take the brain networks into consideration (*Lao et al.*, 2004). Support vector machine (SVM) (*Vapnik*, 1995) is one popular classification method that maps the whole brain into a feature space and then finds a hyperplane in the feature space to separate the two groups. The feature space is determined by a kernel function which needs to be defined before the analysis. Selecting the right kernel is critical to the performance the SVM classifier.

A new method called multiple kernel learning (MKL) has been proposed to combine different kernels together, relaxing the constraint of a single kernel of SVM (*Gönen and Alpaydin*, 2011). Different kernels can represent different similarity measures or can represent different data. The final kernel is a linear combination of several sub-kernels. The human brain consists of functional regions which may contribute to the classifier in different ways. The MKL method can design different kernels on those functional regions and find the best combination of the local kernels.

In this section, we focus on the mathematical formulation of single kernel SVM and MKL SVM methods. We compare the primal form and the dual form of both methods. Section 3.3.1 and section 3.3.2 introduce the SVM and MKL SVM separately. Section 3.3.3 uses a two variables example to explain the similarities and differences of these two methods.

3.3.1 Traditional SVM

SVM was proposed by *Vapnik* (1995) and since then achieved great successes in many fields, such as engineering, geometry and biology. It becomes one of the most popular classifiers in empirical applications. In a classification problem, we are given n subjects. Let pair $(\overrightarrow{\mathbf{x}_i}, y_i)$ be the data of the i-th subject. $\overrightarrow{\mathbf{x}_i} \in \mathbb{R}^V$ represents all the voxels in the gray matter that are used in the analysis. y_i is the class label for the

subject i, with $y_i = 1$ indicating a patient and $y_i = -1$ indicating a healthy control. The goal here is to come up with a decision rule D which is a function from the space of x_i to the space of y_i, $D : \mathbb{R}^V \to \{1, -1\}$. Any new subject can be classified to one of the two groups using function D.

The SVM method approaches this problem by finding a decision hyperplane in the feature space. Let g be the feature function that maps the original signal $\overrightarrow{\mathbf{x}}$ to a feature space with dimension P, $g : \mathbb{R}^V \to \mathbb{R}^P$. Let $\overrightarrow{\mathbf{z}}_i$ be the feature of data $\overrightarrow{\mathbf{x}}_i$ coming from the feature function g, $\overrightarrow{\mathbf{z}}_i = g(\overrightarrow{\mathbf{x}}_i)$. The kernel function H is the Euclidean inner product between the features of two subjects in the feature space, $H(\overrightarrow{\mathbf{x}}_i, \overrightarrow{\mathbf{x}}_j) = \langle \overrightarrow{\mathbf{z}}_i, \overrightarrow{\mathbf{z}}_j \rangle = \langle g(\overrightarrow{\mathbf{x}_i}), g(\overrightarrow{\mathbf{x}_j}) \rangle$. The decision boundary d is a hyperplane in the feature space, which has the following form:

$$d(\overrightarrow{\mathbf{z}}) = \langle \overrightarrow{\mathbf{w}}, \overrightarrow{\mathbf{z}} \rangle + w_0,$$

where $\overrightarrow{\mathbf{w}}$, w_0 are the parameters of the hyperplane. The decision function D takes the sign of the decision boundary plane.

$$D(\overrightarrow{\mathbf{x}}) = \text{sign}(d(\overrightarrow{\mathbf{z}})).$$

This means the subjects on the same side of the decision boundary will be classified in the same class by function D.

SVM finds the decision hyperplane d that can separate the two classes as far as possible. In order to satisfy that, parameters $\overrightarrow{\mathbf{w}}$ and w_0 need to be the solution of the following optimization problem

$$\begin{aligned} \min_{\overrightarrow{\mathbf{w}}, w_0} \quad & \tfrac{1}{2}\|\overrightarrow{\mathbf{w}}\|^2 + C \textstyle\sum_i \xi_i \\ \text{subject to} \quad & y_i(\langle \overrightarrow{\mathbf{w}}, \overrightarrow{\mathbf{z}}_i \rangle + w_0) \geq 1 - \xi_i \\ & \xi_i \geq 0, \quad i = 1, \ldots, n \quad . \end{aligned} \tag{3.1}$$

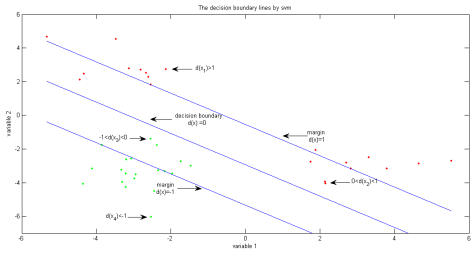

Figure 3.1: The decision boundary and margins of SVM classifier

.

Let margin be the lines satisfying the equation $|d(\overrightarrow{\mathbf{z}})| = 1$. Minimizing the term $\|\overrightarrow{\mathbf{w}}\|^2$ maximizes the distance between the decision boundary and the margin. ξ_i is the slack variable which measures the degree of misclassification of data $\overrightarrow{\mathbf{x}}_i$. If data $\overrightarrow{\mathbf{x}}_i$ is on the correct side of the decision boundary outside or on the margin, then $|\langle \overrightarrow{\mathbf{w}}, \overrightarrow{\mathbf{z}}_i \rangle + w_0| \geq 1$ and $\langle \overrightarrow{\mathbf{w}}, \overrightarrow{\mathbf{z}}_i \rangle + w_0$ has the same sign as y_i which gives $\xi_i = 0$. If data $\overrightarrow{\mathbf{x}_i}$ is on the correct side of the decision boundary but within the margin, then $|\langle \overrightarrow{\mathbf{w}}, \overrightarrow{\mathbf{z}_i} \rangle + w_0| < 1$ and $\langle \overrightarrow{\mathbf{w}}, \overrightarrow{\mathbf{z}_i} \rangle + w_0$ has the same sign as y_i which leads to $0 < \xi_i < 1$. If data $\overrightarrow{x_i}$ is on the wrong side of the decision boundary, then $\langle \overrightarrow{\mathbf{w}}, \overrightarrow{\mathbf{z}_i} \rangle + w_0$ has the opposite sign of y_i, $\xi_i > 1$. So minimizing $\sum_i \xi_i$ controls the misclassification error of the classifier. C is a tuning parameter controlling the trade-off between the complexity of the decision boundary and the training accuracy. If we put a large C, we will have a classifier that has a good performance on the training set but can be over-fitting and not performing well on the testing set. So SVM minimizes the combination of distance between the margin and the decision boundary and the training misclassification error. Figure 3.1 shows the decision boundary and margins of SVM classifier in the feature space. The red dots represent patient and the green dots represent patients. SVM achieves a good classifier between these two groups.

The optimization problem (3.1) is called the primal problem of SVM which has a corresponding dual problem that is easier to solve. The dual optimization problem

has the following form:

$$\begin{aligned} \max_{\{\alpha_i\}_{i=1}^n,} \quad & \textstyle\sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i\alpha_j y_i y_j H(\overrightarrow{\mathbf{x}_i}, \overrightarrow{\mathbf{x}_j}) \\ \text{subject to} \quad & H(\overrightarrow{\mathbf{x}_i}, \overrightarrow{\mathbf{x}_j}) = \langle \overrightarrow{\mathbf{z}_i}, \overrightarrow{\mathbf{z}_j} \rangle \qquad i,j = 1,\ldots,n \\ & \textstyle\sum_i \alpha_i y_i = 0, \quad 0 \le \alpha_i \le C \qquad i = 1,\ldots,n. \end{aligned} \tag{3.2}$$

The dual problem of SVM is a minimization problem over a set of parameters $\{\alpha_i\}_{i=1}^n$. The parameters $\{\alpha_i\}_{i=1}^n$ have a close relationship with the decision hyperplane, such that $\overrightarrow{\mathbf{w}} = \sum_i \alpha_i y_i \overrightarrow{\mathbf{z}}_i$. The decision boundary can be written in $\{\alpha_i\}_{i=1}^n$:

$$\begin{aligned} d(\overrightarrow{\mathbf{z}}) &= \langle \overrightarrow{\mathbf{w}}, \overrightarrow{\mathbf{z}} \rangle + w_0 \\ &= \sum_i \alpha_i y_i \langle \overrightarrow{\mathbf{z}}_i, \overrightarrow{\mathbf{z}} \rangle + w_0. \end{aligned}$$

There are several algorithms proposed to solve (3.2). Because of the one-to-one map between the primal and dual problem, the solution of the primal problem (3.1) can be easily found by the solution of the dual problem (3.2).

3.3.2 Multiple Kernel Learning SVM

The multiple kernel learning (MKL) SVM (*Lanckriet et al.*, 2004) is similar to the traditional SVM problem except it uses multiple kernels other than a single one. In multiple kernel analysis, there are M feature functions $g_1, \ldots, g_M$, each mapping from the original space of $\overrightarrow{\mathbf{x}}$ to a feature space. Let $\overrightarrow{\mathbf{z}}_i^m \in \mathbb{R}^{V_m}$ be the m-th feature of the i-th data from function g_m, $\overrightarrow{\mathbf{z}}_i^m = g_m(\overrightarrow{\mathbf{x}}_i)$. Then the m-th kernel function H_m is defined as $H_m(\overrightarrow{\mathbf{x}}_i, \overrightarrow{\mathbf{x}}_j) = \langle \overrightarrow{\mathbf{z}}_i^m, \overrightarrow{\mathbf{z}}_j^m \rangle$. MKL SVM finds the best kernel H as a linear combination of M kernels. $H(\overrightarrow{\mathbf{x}_i}, \overrightarrow{\mathbf{x}_j}) = \sum_{m=1}^M \eta_m H_m(\overrightarrow{\mathbf{x}_i}, \overrightarrow{\mathbf{x}_j})$ with the constraint that $\sum_{m=1}^M \eta_m = 1, \quad \eta_m \ge 0, \quad m = 1,\ldots,M$. $\{\eta_m\}_{m=1}^M$ are the kernel weights that will be automatically learned by the MKL algorithm. The dual problem of the MKL SVM

is the following optimization problem:

$$\begin{aligned} \min_{\{\eta_m\}_{m=1}^M} \max_{\{\alpha_i\}_{i=1}^n} \quad & \textstyle\sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i\alpha_j y_i y_j H(\overrightarrow{\mathbf{x}}_i, \overrightarrow{\mathbf{x}}_j) \\ \text{subject to} \quad & H = \textstyle\sum_{m=1}^M \eta_m H_m, \quad H_m(\overrightarrow{\mathbf{x}}_i, \overrightarrow{\mathbf{x}}_j) = \langle \overrightarrow{\mathbf{z}}_i, \overrightarrow{\mathbf{z}}_j \rangle \\ & \textstyle\sum_m \eta_m = 1, \quad \eta_m \geq 0, \quad \sum_i \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C \\ & i, j = 1, \ldots, n, \quad m = 1, \ldots, M, \end{aligned} \tag{3.3}$$

The dual form of MKL SVM is similar to the dual form of traditional SVM except it replaces the given kernel H by a weighted sum of M kernels $H_1, \ldots, H_M$. The algorithm finds the best weights $\{\eta_m\}_{m=1}^M$ that minimize the maximization problem of the traditional SVM dual. MKL SVM searches the best classifier in a larger space which brings more flexibility to the classifier and usually leads to a better performance.

The decision boundary of the MKL SVM also shares a very close relationship to the form of decision boundary in traditional SVM. Let $\overrightarrow{\mathbf{w}}_m = \sum_i \alpha_i y_i \overrightarrow{\mathbf{z}}_i^m$ be the parameters of the individual decision boundary in the m-th kernel space which takes the form of boundary parameters in the traditional SVM. It can be shown that the parameters for the MKL SVM boundary is the alignment of parameters of individual boundary with kernel weights, $\overrightarrow{\mathbf{w}} = (\eta_1 \overrightarrow{\mathbf{w}}_1, \ldots, \eta_M \overrightarrow{\mathbf{w}}_M)$. Then for a new subject $\overrightarrow{\mathbf{z}} = (\overrightarrow{\mathbf{z}}^1, \ldots, \overrightarrow{\mathbf{z}}^M) = (g_1(\overrightarrow{\mathbf{x}}), \ldots, g_M(\overrightarrow{\mathbf{x}}))$ the decision boundary is

$$\begin{aligned} d(\overrightarrow{\mathbf{z}}) &= \langle \overrightarrow{\mathbf{w}}, \overrightarrow{\mathbf{z}} \rangle + w_0 \\ &= \langle (\eta_1 \overrightarrow{\mathbf{w}}_1, \ldots, \eta_M \overrightarrow{\mathbf{w}}_M), (\overrightarrow{\mathbf{z}}^1, \ldots, \overrightarrow{\mathbf{z}}^M) \rangle + w_0 \\ &= \sum_{m=1}^M \eta_m \langle \overrightarrow{\mathbf{w}}_m, \overrightarrow{\mathbf{z}}^m \rangle + w_0 \\ &= \sum_{m=1}^M \eta_m \sum_i \alpha_i y_i \langle \overrightarrow{\mathbf{z}}_i^m, \overrightarrow{\mathbf{z}}^m \rangle + w_0. \end{aligned}$$

$\overrightarrow{\mathbf{w}}_m$ defines a hyperplane in the m-th feature space. η_m is the kernel weight representing the contribution of the m-th kernel to the decision boundary. The kernel weights of the kernels with no information about the class label will be set to 0. This can be seen from the primal problem of the MKL:

$$\begin{aligned} \min_{\{\overrightarrow{\mathbf{w}_m}\}_{m=1}^M,\xi} \quad & \tfrac{1}{2}(\textstyle\sum_{m=1}^M \|\overrightarrow{\mathbf{w}}_m\|_2)^2 + C\sum_{i=1}^N \xi_i \\ \text{subject to} \quad & y_i(\textstyle\sum_m \langle \overrightarrow{\mathbf{w}}_m, \overrightarrow{\mathbf{z}}_i^m \rangle + w_0) \geq 1 - \xi_i \quad \overrightarrow{\mathbf{w}_m} \in \mathbb{R}^{V_m}, \quad m = 1, \ldots, M \\ & \overrightarrow{\mathbf{z}}_i^m = g_m(\overrightarrow{\mathbf{x}}_i), \quad \xi_i \geq 0 \qquad i = 1, \ldots, n. \end{aligned} \tag{3.4}$$

Optimization problem (3.4) is the primal form of MKL SVM. The optimization problem minimizes a weighted sum of the penalty term $\frac{1}{2}(\sum_{m=1}^M \|\overrightarrow{\mathbf{w}}_m\|_2)^2$ and the training misclassification error $\sum_{i=1}^N \xi_i$. The penalty term has a blocked l_1 norm which means within each kernel, it penalizes the l_2 norm of the boundary parameter $\overrightarrow{\mathbf{w}}_m$ and among different kernels it penalizes the l_1 norm which is a linear sum of all the l_2 norms. It is well-known that l_1 norm penalty has a nice sparsity property, which means $\eta_m = 0$ for some $m \in \{1, \ldots, M\}$. If feature m is informative about the class label, η_m will be strictly positive. If feature m is not informative about the class label, η_m will be set to 0 which means the final classifier will not use any information in the kernel m. This sparsity property can be used as a feature selection tool which will increase the performance of classifier and also identify the informative local regions for structural MRI data.

Both the traditional SVM and the MKL SVM are multivariate approaches since both decision boundaries contain the information from multiple variables. There is a list of similarities between the two methods. We put the notation and the formula of SVM and MKL SVM in Table 3.1 for comparison.

The difference is that MKL SVM combines the information from different variables

	SVM		Multi-SVM	
	Notation	Formula	Notation	Formula
Data	$(\overrightarrow{\mathbf{x}}_i, y_i)$		$(\overrightarrow{\mathbf{x}}_i, y_i)$	
Features Functions	g		g	$g = (g_1, \ldots, g_M)$
Features	$\overrightarrow{\mathbf{z}}_i$	$\overrightarrow{\mathbf{z}}_i = g(\overrightarrow{\mathbf{x}}_i)$	$\overrightarrow{\mathbf{z}}_i$	$\overrightarrow{\mathbf{z}}_i = (\overrightarrow{\mathbf{z}}_i^1, \ldots, \overrightarrow{\mathbf{z}}_i^M),\ \overrightarrow{\mathbf{z}}_i^m = g_m(\overrightarrow{\mathbf{x}}_i)$
Kernel Function	H	$H(\overrightarrow{\mathbf{x}}_i, \overrightarrow{\mathbf{x}}_j) = \langle \overrightarrow{\mathbf{z}}_i, \overrightarrow{\mathbf{z}}_j \rangle$	H	$H = \sum_m \eta_m H_m,\ H_m(\overrightarrow{\mathbf{x}}_i, \overrightarrow{\mathbf{x}}_j) = \langle \overrightarrow{\mathbf{z}}_i^m, \overrightarrow{\mathbf{z}}_j^m \rangle$
Kernel Weights			$(\eta_i)_{i=1}^n$	
Parameters	$\overrightarrow{\mathbf{w}},\ w_0$	$\overrightarrow{\mathbf{w}} = \sum_i \alpha_i y_i \overrightarrow{\mathbf{z}}_i$	$\overrightarrow{\mathbf{w}},\ w_0$	$\overrightarrow{\mathbf{w}} = (\overrightarrow{\mathbf{w}}^1, \ldots, \overrightarrow{\mathbf{w}}^M),\ \overrightarrow{\mathbf{w}}^m = \eta_m * \sum_i \alpha_i y_i \overrightarrow{\mathbf{z}}_i^m$
Decision Boundary	d	$d(\overrightarrow{\mathbf{z}}) = \langle \overrightarrow{\mathbf{w}}, \overrightarrow{\mathbf{z}} \rangle + w_0$	d	$d(\overrightarrow{\mathbf{z}}) = \sum_{m=1}^M \eta_m \langle \overrightarrow{\mathbf{w}}_m, \overrightarrow{\mathbf{z}}^m \rangle + w_0$
Primal Problem	$\min_{\overrightarrow{\mathbf{w}}, w_0} \frac{1}{2}\|\overrightarrow{\mathbf{w}}\|^2 + C\sum_i \xi_i$ subject to $y_i(\langle \overrightarrow{\mathbf{w}}, \overrightarrow{\mathbf{z}}_i \rangle + w_0) \geq 1 - \xi_i$ $\xi_i \geq 0 \quad i = 1, \ldots, n$		$\min_{\overrightarrow{\mathbf{w}_m}, w_0, \xi} \frac{1}{2}(\sum_{m=1}^M \|\eta_m \overrightarrow{\mathbf{w}}_m\|_2)^2 + C\sum_{i=1}^N \xi_i$ subject to $y_i(\sum_m \eta_m \langle \overrightarrow{\mathbf{w}}_m, \overrightarrow{\mathbf{z}}_i^m \rangle + w_0) \geq 1 - \xi_i$ $\xi_i \geq 0 \quad i = 1, \ldots, n$	
Dual Problem	$\max_{\alpha_i} \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j H(\overrightarrow{\mathbf{x}_i}, \overrightarrow{\mathbf{x}_j})$ $\sum_i \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C$		$\min_{\eta_m} \max_{\alpha_i} \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j H(\overrightarrow{\mathbf{x}}_i, \overrightarrow{\mathbf{x}}_j)$ $\sum_m \eta_m = 1, \quad \eta_m \geq 0, \quad \sum_i \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C$	

Table 3.1: Formula for SVM and MKL-SVM

at the kernel levels through the kernel weights while the traditional SVM combines the information from different variables at the feature levels through the original data. The benefit for combining at the kernel level is that it allows the algorithm to search the optimum classifier over different kernels which may capture different structures of the data. Moreover, the kernel weights of the uninformative kernels will be set to 0 in MKL case. So MKL can distinguish the informative features from uninformative ones more efficiently than the traditional SVM. This suits for the high dimensional data with only few informative variables and lots of noises. This sparsity property is one of the main reasons that the MKL SVM usually outperforms the traditional SVM in the analysis of brain image data.

3.3.3 Toy Example

We use a toy example to illustrate the relation between MKL classifier and its sub-classifiers based on individual kernels. We generate 40 data $\{(X_i, Y_i)\}_{i=1}^{40}$ where subjects $1, \ldots, 20$ are patients with $Y_i = 1$ and subjects $21, \ldots, 40$ are healthy controls with $Y_i = -1$. There are two variables $X_i = \{x_{i1}, x_{i2}\}$. For subject $i = \{1, \ldots, 10\}$, the first variable x_{i1} follows a normal distribution with mean 3 and standard deviation 1 and the second variable x_{i2} follows a normal distribution with mean -3 and standard deviation 1. These are the red dots in the lower right of figure 3.2 and figure 3.3. For subject $i = \{11, \ldots, 20\}$, the first variable x_{i1} follows a normal distribution with mean -3 and standard deviation 1 and the second variable x_{i2} follows a normal distribution with mean 3 and standard deviation 1. These are the red dots in the upper left of figure 3.2 and figure 3.3. For subject $i = \{21, \ldots, 40\}$ in the control group, both variables follow a normal distribution with mean -3 and standard deviation 1. These are the green dots in the figure 3.2 and figure 3.3. In this setting, each variable can only distinguish 10 patients from the healthy controls. So if we design one linear kernel on a single variable and train a SVM classifier based on that individual variable, we

can not get a good performance. The second and the third figures in figure 3.2 show the classifier of the SVM on the first and the second variable separately. The decision boundary for SVM on the first variable is $0.3753 * x_1 + 0.1990 = 0$ which is the middle line in the second figure in figure 3.2. The right and left lines are $0.3753 * x_1 + 0.1990 = \pm 1$ which are the margins. The decision boundary for SVM on the second variable is $0.3591 * x_2 + 0.2371 = 0$ which is the middle line in the third figure in figure 3.3. The right and left lines are the margins. The classifier based on first variable only distinguishes the subject $i = 1, \ldots, 10$ while the classifier based on second variable only distinguishes the subject $i = 11, \ldots, 20$. So SVM on an individual variable does not perform well. But if we combine the two kernels together, the MKL algorithm will give none zero weights to both kernels and find the best classifier as a linear combination of both sub-classifiers. In this way, the MKL can not information from both variables. The figure 3.3 shows the result of MKL SVM. The first figure and the second figure show the sub-classifiers based on the first and the second variable by MKL SVM separately. The parameter of the decision plane in the space of the first variable is $w_1 = 0.8118$ and the parameter of the decision plane in the space of the second variable is $w_2 = 0.7962$. The intercept $w_0 = 1.2252$. So the decision boundary for the first sub-kernel is $d_1(x_1) = w_1 * x_1 + w_0 = 0.8118 * x_1 + 1.2252 = 0$ and the decision boundary for the second sub-kernel is $d_2(x_2) = w_2 * x_2 + w_0 = 0.7965 * x + 1.2252 = 0$. We can see individually, these sub-classifiers from the MKL SVM do not perform better than the SVM on an individual variables. However, the multiple kernel classifier combining the two sub-classifiers can achieve a clean separation of the two groups. The weights of the first sub-classifiers are $\eta_1 = 0.4767$ and $\eta_2 = 0.5233$. So the final decision plane is $\eta_1 * d_1(\overrightarrow{x}_1) + \eta_2 * d_2(\overrightarrow{x}_2) + b = 0.4767 * 0.8118 * \overrightarrow{x}_1 + 0.5233 * 0.7965 * \overrightarrow{x}_2 + 1.2252 = 0$. This is the blue line in the middle the third figure in figure 3.3. So in this toy example, we can see the if individual variables only hold part information about the structure of

the data, MKL can gather the information together by combining the sub-classifiers based on individual variables. It can learn the kernel weights efficiently to get a classification performance.

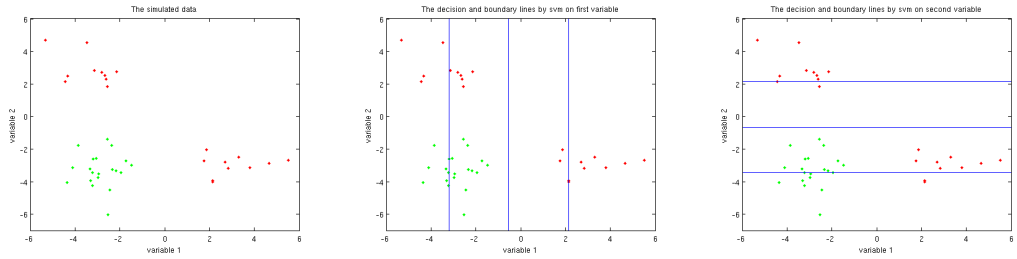

Figure 3.2: Decision boundaries for single kernel SVM

Figure 3.2 shows the decision boundaries for single kernel SVM. The first figure is the simulated data. The red dots are the patients and the green dots are the healthy controls. The patients split into two sub-groups, one is in the upper left and another is in the right bottom of the figure. The second figure and the third figure show the classifier by SVM on the first variable and the second variable separately. The blue lines in the middle from both figures are the decision boundary by SVM on each variable. The right and left lines are the margins for different groups.

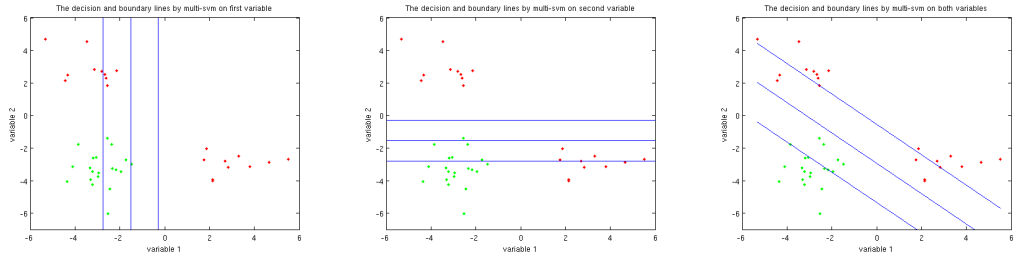

Figure 3.3: Decision boundaries for MKL

Figure 3.3 shows the decision boundary for MKL SVM. The first figure is the sub-classifier of the first variable by MKL SVM. The second figure is the sub-classifier of the second of the second variable by MKL SVM. The third figure is the final classifier on both variables by multiple kernel SVM. The final classifier by MKL is a linear combination of different kernels, in this case different variables. In this toy

example, each kernel was build on an individual variable, both the MKL SVM and traditional SVM use a straight line in a two dimension space to separate the two groups. However, in high dimension problem, especially when individual kernel was build on several variables, these two methods will show significant difference because of the sparsity property of the multiple kernel algorithm.

3.4 Simulation

In this section, we test the two-step procedure on simulated data sets. Taking both the localization and integration properties of the brain, we design several localized informative regions scattered among some noninformative regions on a two dimensional image. Informative regions are the ones can distinguish the patients from the healthy controls while the noninformative regions can not. We put four distant information regions to see firstly if the multiple kernel learning can pick the significant ones from the others and further gain the strength by combining the information regions together. Also based on the real data, we add spatial correlation in both the informative and noninformative regions and test how different levels of correlation can influence the multiple kernel learning results. Another important issue is that the brain image data is notorious for its lower signal to noise ratio. So different levels of white noises are tested for multiple kernel learning. We compare the classification error of the two-step procedure to the individual kernel learning in different settings to see how different parameters influence the performance.

3.4.1 Simulation Framework

We generate our data as a 50×50 image. Let $\mu^{pat} \in \mathbb{R}^{50\times 50}$ represent the mean image of the patient group and $\mu^{con} \in \mathbb{R}^{50\times 50}$ represent the mean image of the control group. The mean image is designed to be a weighted sum of a background image which is the same for both groups and a informative image which is different for two

groups. The informative image has nonzero values on the informative regions and zero everywhere else. Let $\mu_{back} \in \mathbb{R}^{50\times 50}$, $\mu_{inf}^{pat} \in \mathbb{R}^{50\times 50}$ and $\mu_{inf}^{con} \in \mathbb{R}^{50\times 50}$ represent the background image, the informative image of the patient group and the informative image of the control group separately. Then the mean image can be expressed as following:

$$\mu^{pat} = (\mathbf{I} - \omega) * \mu_{back} + \omega * \mu_{inf}^{pat},$$

$$\mu^{con} = (\mathbf{I} - \omega) * \mu_{back} + \omega * \mu_{inf}^{con},$$

where $\mathbf{I}$ is a 50×50 matrix with all entries 1.

The background image μ_{back} is generated from a Gaussian field with mean 0:

$$\mu_{back} \sim \mathcal{GF}(0, \Sigma_{back}).$$

Σ_{back} is a 250×250 covariance matrix. For pixels P_{i_1,j_1} and P_{i_2,j_2} ($\{i_1, j_1, i_2, j_2\} \subset \{1, \ldots, 50\}$), the correlation $\Sigma_{back}(P_{i_1,j_1}, P_{i_2,j_2})$ is a function of the distance between them:

$$\Sigma_{back}((i_1, j_1), (i_2, j_2)) = \exp(-\sqrt{(i_1 - i_2)^2 + (j_1 - j_2)^2}/C_{back}),$$

where C_{back} is the parameter controlling the level of correlation in the background image. As C_{back} decreases, the correlation levels in the background image decreases exponentially.

The informative images μ_{inf}^{pat} and μ_{inf}^{con} contain the group information with nonzero values only on the informative regions and zero everywhere else. The size and the location of the informative regions are fixed, shown in Figure 3.4. The nonzero values within each informative regions are generated from Gaussian fields. Let $\mu_{inf,t}^{pat}$ and $\mu_{inf,t}^{con}$ be one of the four informative regions in the mean image for patient and control

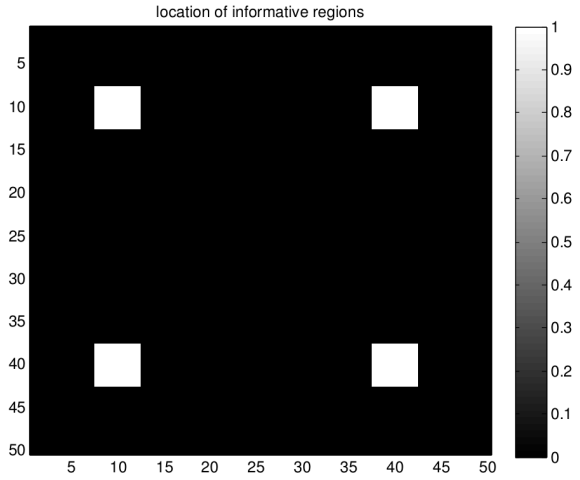

Figure 3.4: The location of informative regions in the mean image. The white regions near four corners are the informative regions. The black regions are the noninformative regions.

groups, each region is generated separately:

$$\mu_{inf,t}^{pat} \sim \mathcal{GF}(\mu_0 * \mathbf{I}, \sigma_{inf}^2 * \Sigma_{inf}),$$

$$\mu_{inf,t}^{con} \sim \mathcal{GF}(-\mu_0 * \mathbf{I}, \sigma_{inf}^2 * \Sigma_{inf}),$$

where $\mathbf{I}$ is a vector with all entries 1. μ_0 is the mean value of the Gaussian field used to generated the informative regions. σ_{inf}^2 is the level of fluctuation across different pixels in the informative regions. Σ_{inf} is the correlation matrix of the informative regions. The correlation between two pixels in the informative regions is also defined as a function of the distance:

$$\Sigma_{inf}((i_1, j_1), (i_2, j_2)) = \exp(-\sqrt{(i_1 - i_2)^2 + (j_1 - j_2)^2}/C_{inf}),$$

where C_{inf} controls the level of correlation in the informative regions.

$\omega \in \mathbb{R}^{50\times 50}$ is the weight matrix, controlling the proportion of the contribution from the background image and the informative images. To get a smooth boundary of the informative regions, we use a two dimension parabolic function for ω as shown

in the left figure of Figure 3.5. The right figure of Figure 3.5 shows the weight image ω.

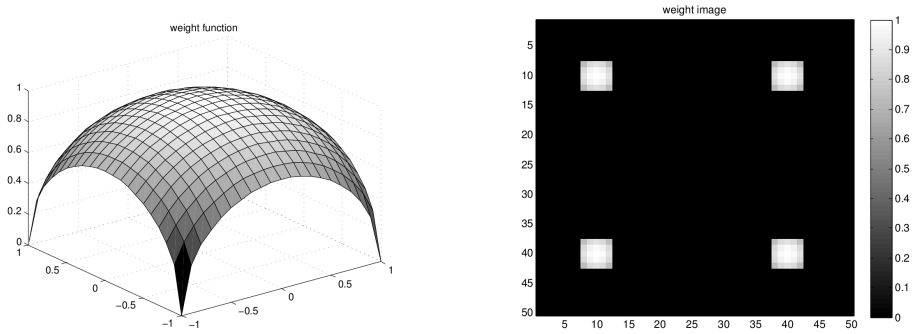

Figure 3.5: Weight function and weight image of ω. Left: the parabolic weight function. Right: the weight image ω. ω is close to 1 in the center of the informative regions and decrease as approaching the boundary.

Let X^{pat} and X^{con} represent the data of the patient group and the control group. Each observation is the mean image of the corresponding group plus white noise ϵ.

$$X^{pat} = \mu^{pat} + \epsilon,$$

$$X^{con} = \mu^{con} + \epsilon,$$

where $\epsilon \in \mathbb{R}^{50\times 50}$ is the noise image, $\epsilon_{i,j} \sim \mathcal{N}(0, \sigma^2_{noi})$ is independent across different subjects and different pixels. Parameter σ^2_{noise} controls the level of white noise.

In the framework described above, there are five parameters in the model to simulate the data. Parameters μ_0, σ_{inf} and σ_{noise} control the signal to noise ratio in the image. Parameters C_{inf} and C_{back} control the correlation level in the image. We try different values for these parameters to see how they change the classification error rate.

3.4.2 Two-Step Procedure

30 50×50 images from each group are generated to fit a two-step multiple kernel learning classifier. A total of 60 images are split into 36 training images, 12 validation

images and 12 test images. Every 50×50 image is divided into 100 non-overlapped 5×5 regions. Each informative regions have overlapped pixels with four neighboring regions. For each generated data set, $\{x_1^{con}, \ldots, x_{30}^{con}, x_1^{pat}, \ldots, x_{30}^{pat}\}$, we leave 12 subjects aside as test data. And then the rest are split into 12 validation data and 36 training data. A single kernel SVM is applied to each individual region on the training set and test on the validation data. This validation and test splitting are repeated 50 times to get a mean validation error rate for each individual region. At the second step, we apply a multiple kernel SVM on the top K regions with the lowest validation error to get a final classifier. This classifier is trained on the training and validation data and then test on the test data. This testing splits are also repeated 50 times to get a mean test error rate for the multiple kernel classifier. We test the multiple kernel classifier under different values of K. And finally we compare the multiple kernel classifier to the best individual region in terms of their test errors to see if combining several regions can outperform the individual classifier. The procedure is put in a flow chart in Figure 3.6.

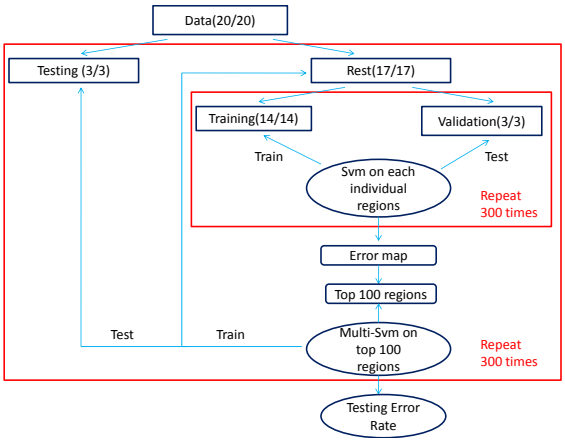

Figure 3.6: The float chart of the method. 60 subjects are divided into test set (12 subjects), training set (36 subjects) and validation set (12 subjects). At the first step, a single kernel SVM is trained on each region of the training set and tested on the validation set. The training and validation splits are repeated 50 times to get an average validation error rate for each region. At the second step, a multiple kernel learning is applied to the top K regions with the lowest validation errors to get a final classification error rate. The multipel kernel classifier is trained on the training and validation sets together and then test on the test sets. The test set splits are repeated 50 times to get an average error rate for the multiple kernel learning.

3.4.3 Result

This section shows the results of multiple kernel learning in different scenarios. We exam how each parameter can influence the test error of the multiple kernel classifier and compare the results to the single kernel classifier.

1. σ_{noise}

   σ_{noise} is the standard deviation of white noise ϵ. The larger the σ_{noise}, the smaller the signal to noise ratio. Figure 3.7 shows the data images of both patient and healthy control groups under different values of σ_{noise}. From Figure 3.7, we can see that μ^{con} and μ^{pat} are smooth images with different values in the informative regions and same values any where else. When $\sigma_{noise} = 0.1$, the data images are almost the same as mean images. When σ_{noise} goes up to 2, the data images are like random noises and the difference in the informative regions are masked by white noises. All the other parameters used to generated the data are fixed $\mu_0 = 0.05$, $\sigma_{inf} = 0.05$, $C_{back} = 2$ and $C_{inf} = 1$.

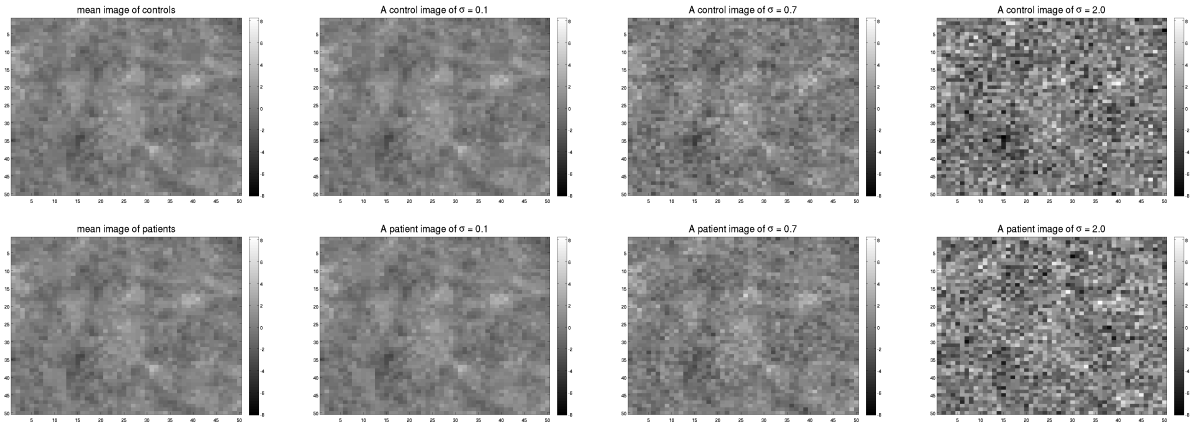

Figure 3.7: The mean images and the data image of different σ_{noise}. From left to right, $\sigma_{noise} = 0$, 0.1, 0.7 and 2. The images in the first row are healthy controls and the lower rows are patients.

   The increase in the σ_{noise} has a direct effect on the misclassification error of single kernel classifiers. Table 3.2 shows the validation error of the best 5 regions

for different values of σ_{noise}. For $\sigma_{noise} = 0.1$, the best individual region can provide enough information about the two groups and achieve an error rate lower than 0.01. When $\sigma_{noise} = 2$, the best individual region can only get a classifier with validation error around 0.35. So as σ_{noise} increases, the informative regions are masked by the white noises and behave more like noninformative regions.

σ_{noise}	1	2	3	4	5
0.1	0.0001	0.0001	0.0001	0.0001	0.0001
0.5	0.18262	0.2214	0.24825	0.26758	0.29445
0.7	0.2727	0.29957	0.32458	0.34242	0.35642
1	0.33575	0.3491	0.36653	0.37777	0.38767
2	0.35413	0.37878	0.38788	0.39303	0.40485

Table 3.2: The validation error of the 5 best regions for different σ_{noise}

We then train the multiple kernel learning classifier on the best regions selected in the single kernel classification step. Figure 3.8 plots the results of multiple kernel learning against different values of σ_{noise}. As σ_{noise} increases, the error rate also increases for all the multiple kernel classifiers. For $\sigma_{noise} = 0.1$, the multiple kernel classifier can achieve a perfect split with 0 misclassification error. For $\sigma_{noise} = 2$, the multiple kernel classifier is like random guess with an error rate around 0.5. This is may because single kernel SVM can not choose the right regions for the multiple kernel learning or simply that the multiple kernel learning can not gain much strength when signal is too weak comparing to the noise. For a fixed value of σ_{noise}, including more regions gives a better classifier than not including enough. As the number of regions for multiple kernel learning exceeds the number of the informative regions, the performance doesn't get worse. From Figure 3.8, we can see the misclassification error rate for $N = 30$, 50, 70 and 100 are almost the same for different values of σ_{noise}. This shows the multiple kernel can distinguish the informative regions from the noninformative regions and put weights on the right ones.

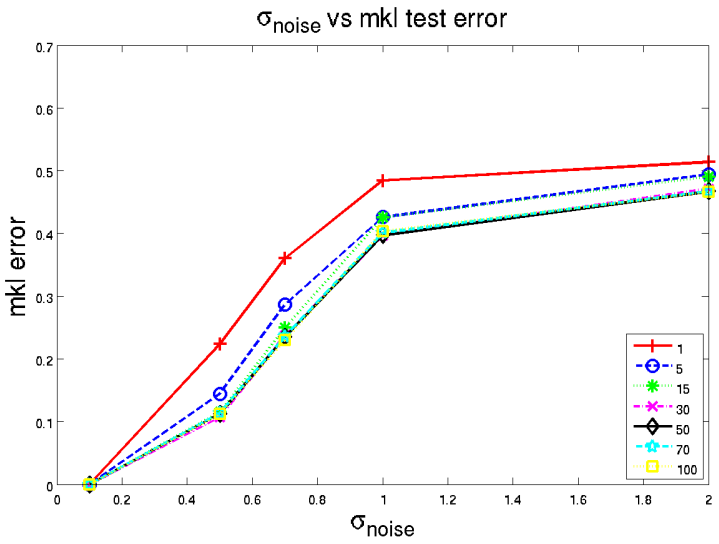

Figure 3.8: Multiple kernel learning results for different σ_{noise}. Different lines in the figures represent different numbers of top regions used in the multiple kernel learning. In the legend from the top to the bottom, the red line, blue line, green line, magenta line, black line, cyan line, and yellow line correspond to $N = 1$, 5, 15, 30, 50, 70 and 100, separately.

Figure 3.9 shows the kernel weights map for multiple kernel learning taking $N = 100$. We can see when $\sigma_{noise} = 0.1$, there exists a clear difference between the informative regions and noninformative regions. The multiple kernel weights are all on the sixteen squares which overlap the informative regions. And when $\sigma_{noise} = 2$, the difference between two types of regions are small comparing to the noise level and the weights are more uniformly distributed for both informative and noninformative regions.

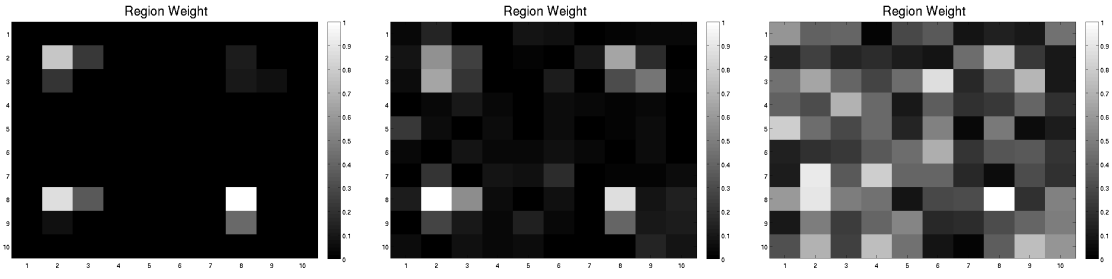

Figure 3.9: Region weight map for different σ_{noise}. From the left to right, the figures are the average region weights for $\sigma_{noise} = 0.1$, 0.7 and 2. All the weights are scaled to be in $[0, 1]$ for each map.

2. μ_0

   μ_0 is the mean of the Gaussian process used to generate the values in the informative regions. The larger the μ_0, the bigger the difference between two groups. Figure 3.10 shows the informative regions of both groups for different values of μ_0. As μ_0 increases, the difference in the informative regions between two groups becomes more distinguishable. When $\mu_0 = 0$, the informative regions of both groups come from the same Gaussian distribution. When $\mu_0 = 0.5$, the difference in the informative regions becomes very obvious. All the other parameters used to generate the data are fixed $\sigma_{noise} = 0.05$, $\sigma_{inf} = 0.05$, $C_{back} = 2$ and $C_{inf} = 1$.

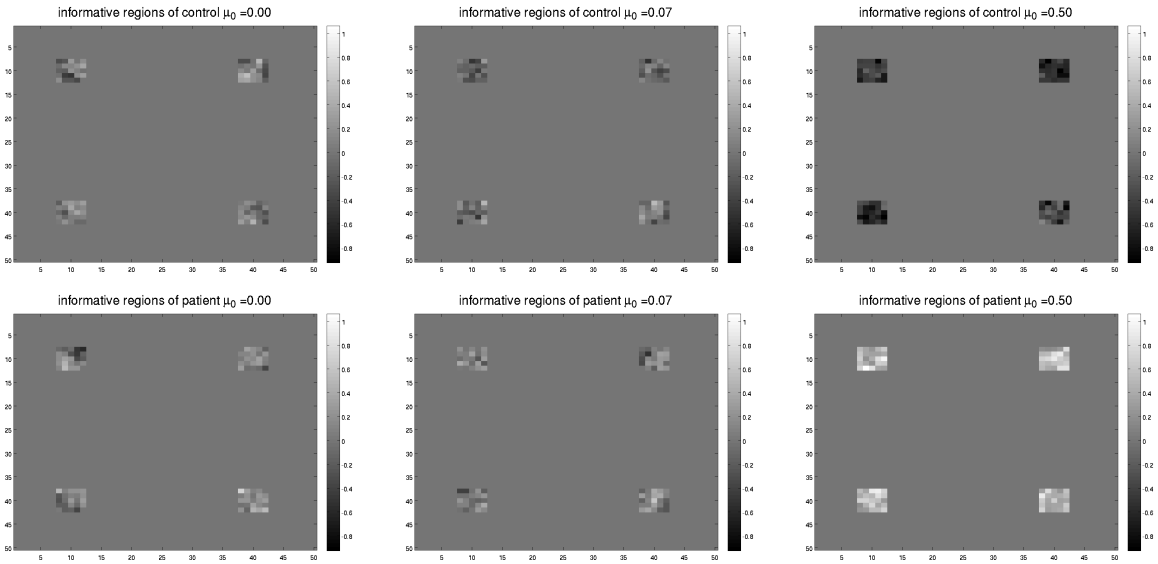

Figure 3.10: Images of the informative regions for different μ_0. From left to right, $\mu_0 = 0$, 0.07 and 0.5. The first row is the control group and the second row is the patient group.

Table 3.3 shows the individual validation error rate of the best 5 regions for different values of μ_0. As μ_0 increases from 0 to 0.5, the best misclassification error rate drops from 0.3 to around 0.016. When $\mu_0 = 0$, the best individual regions perform better than random guess.

Figure 3.11 plots the multiple kernel test errors against the different values of

μ_0	1	2	3	4	5
0	0.29535	0.31522	0.33032	0.34652	0.3572
0.02	0.2845	0.3106	0.33137	0.34795	0.35857
0.07	0.26	0.29342	0.32065	0.34005	0.35658
0.1	0.24293	0.28077	0.31225	0.32565	0.3469
0.5	0.01605	0.034925	0.046775	0.066675	0.07645

Table 3.3: The validation error of the 5 best cubes for different μ_0

μ_0. Different lines in the figure represent different numbers of regions used in the multiple kernel learning. The test error is a decreasing function of μ_0. When $\mu_0 = 0$, the multiple kernel learning error is around 0.42 for $N = 1$ and 0.25 for $N \geq 30$. When $\mu_0 = 0.5$, the test error is 0 no matter how many regions are used. This shows that if single kernels perform well enough, then multiple kernel classifier does not need to include all the informative regions in the second step to get a good performance. When individual regions do not perform well enough, including all the informative regions will result in a lower error rate. Once $N \geq 30$, the error rates stay the same as N increases. This shows when the multiple kernel classifier has all the informative regions, including more noninformative regions won't have much effect on the performance.

Figure 3.12 shows the multiple kernel weights map for different levels of μ_0 for $N = 100$. We can see when $\mu_0 = 0$, the multiple kernel classifier selects some noninformative regions besides the informative regions. As μ_0 increases, the weights are all on the informative regions. This agrees with the misclassification error rate results in Figure 3.11. Less noninformative regions in the final classifier gives a lower error rate.

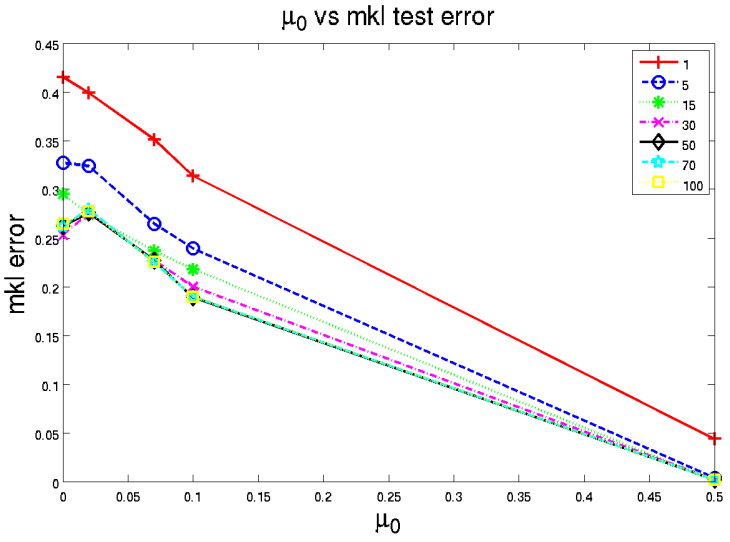

Figure 3.11: Multiple kernel learning results for different μ_0. Different lines in the figures represent different numbers of top regions used in the multiple kernel learning. In the legend from the top to the bottom, the red line, blue line, green line, magenta line, black line, cyan line, and yellow line correspond to $N = 1$, 5, 15, 30, 50, 70 and 100, separately.

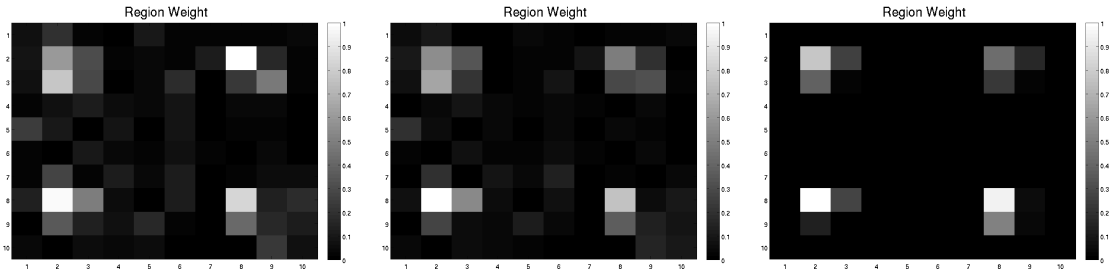

Figure 3.12: Region weight map for different μ_0. From left to right, $\mu_0 = 0$, 0.07 and 0.5 and all the weights are rescaled between 0 and 1.

3. σ_{inf}

σ_{inf} controls the variance level in the informative regions. The larger σ_{inf} gets, the larger the difference between the two groups becomes. Figure 3.13 shows the informative regions of the both groups for different values of σ_{inf}. The figure shows as σ_{inf} increases, the differences between the two groups become more obvious. All other parameters used to generate the data are fixed $\sigma_{noise} = 0.7$, $\mu_0 = 0.05$, $C_{back} = 2$ and $C_{inf} = 1$.

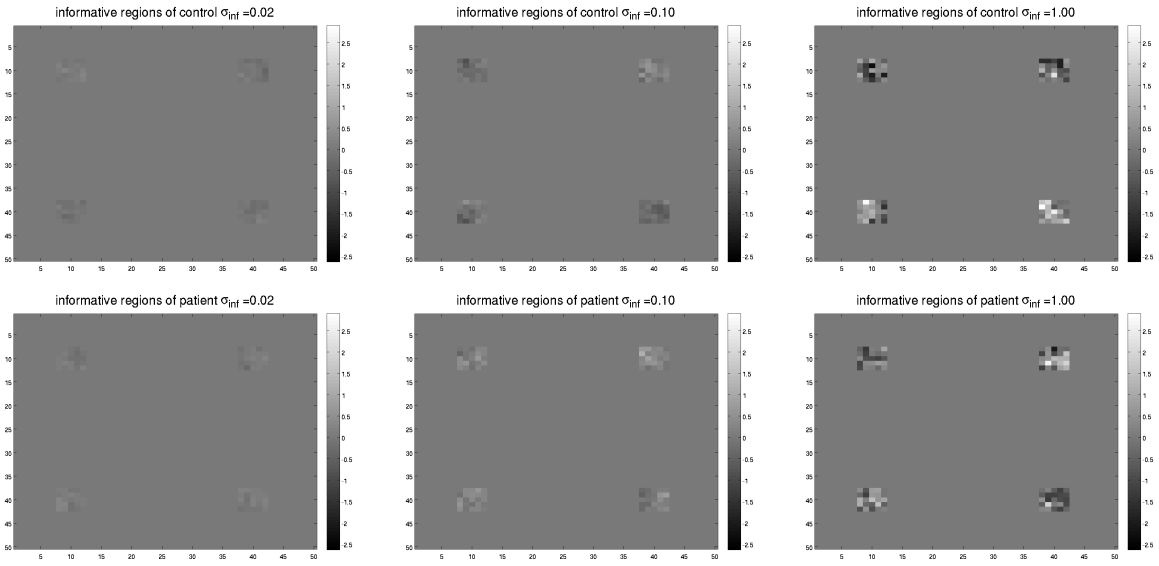

Figure 3.13: Images of the informative regions for different σ_{inf}. The first row are the images of control group and the second row are the images of patient group. From left to right $\sigma_{inf} = 0.02$, 0.1 and 1.

Table 3.4 shows the validation error rate of the 5 best regions for different levels of σ_{inf}. As σ_{inf} increases, the error rate goes down. When σ_{inf} is 0.02, the best individual region gives an error rate around 0.33. As σ_{inf} increases to 1, the best error rate goes down lower than 0.01.

Figure 3.14 shows the results of multiple kernel learning against different values of σ_{inf} for $N = 100$. When σ_{inf} is 0.02, the multiple kernel learning error rates are also very high, around 0.5 for $N = 1$ and 0.4 for $N \geq 30$. The differences among different number of regions are small. As σ_{inf} increases to

σ_{inf}	1	2	3	4	5
0.02	0.33168	0.35352	0.3689	0.37988	0.387
0.05	0.27243	0.30615	0.32195	0.34402	0.3573
0.1	0.19122	0.23598	0.25865	0.28362	0.30625
0.5	0.0171	0.03555	0.05215	0.06735	0.0807
1	0.00025	0.001675	0.006175	0.011675	0.0189

Table 3.4: The validation error of the 5 best regions for different σ_{inf}

0.1, the multiple kernel classifier begins to perform significantly better than the individual one, improving the misclassification error rate by 50% percent. As σ_{inf} goes above 0.5, the multiple kernel classifiers can get almost perfect splits for all values of N. This shows that when the signal is very weak, both individual and multiple kernel classifiers can not perform well. The benefit of combining different kernels is very limited. As the signal to noise ratio becomes higher, the multiple kernel classifier begins to perform significantly better than individual one. And when the signal becomes so strong that the individual kernel classifier can perform well, the misclassification error rates from both single kernel and multiple kernel classifiers converges to 0.

Figure 3.15 compares the weights map for different levels of σ_{inf}. When $\sigma_{inf} = 0.01$, the multiple kernel learning selects some noninformative regions. As σ_{inf} goes up to 1, the weights are only concentrated on a few informative regions.

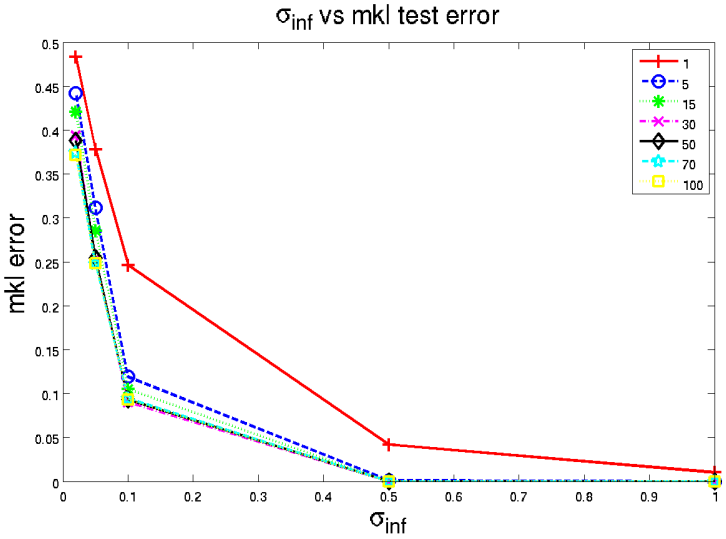

Figure 3.14: Multiple kernel learning results for different σ_{inf}. Different lines in the figures represent different numbers of top regions used in the multiple kernel learning. In the legend from the top to the bottom, the red line, blue line, green line, magenta line, black line, cyan line, and yellow line correspond to $N = 1$, 5, 15, 30, 50, 70 and 100, separately.

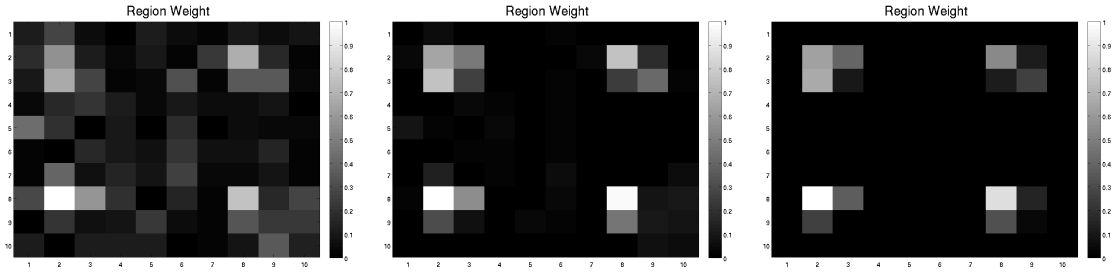

Figure 3.15: Region weight map for different σ_{inf}. From the left to right are weight maps of $\sigma_{inf} = 0.02$, 0.1 and 1 for $N = 100$. All the weights are rescaled between 0 and 1.

4. C_{back}

   C_{back} is the parameter controlling the spatial correlation level in the background image. Figure 3.16 shows the background images for different values of C_{back}. As C_{back} increases, the background images become more smooth with higher spatial correlation. All other parameters used to generate the data are fixed $\sigma_{noise} = 0.7$, $\sigma_{inf} = 0.05$, $\mu_0 = 0.05$ and $C_{inf} = 1$.

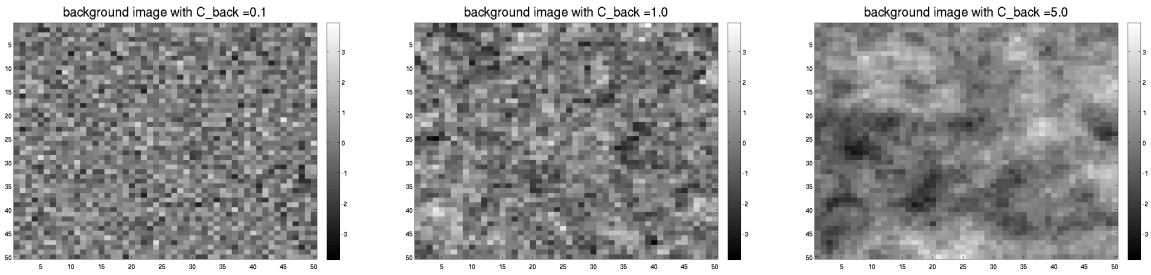

Figure 3.16: Background images for different C_{back}. From left to right $C_{back} = 0.1$, 1 and 5.

Table 3.5 shows the individual validation error rates of the 5 top regions for different values of C_{back}. Since C_{back} controls the correlation level in the background image, it is not directly related to the difference between two groups. Table 3.5 shows for different levels of correlation, the best individual validation error rates are about the same level around 0.27.

C_{back}	1	2	3	4	5
0.1	0.27843	0.3069	0.3241	0.34227	0.35702
0.5	0.2741	0.31152	0.32932	0.33908	0.3543
0.7	0.2737	0.302	0.32475	0.34265	0.35885
1	0.2729	0.305	0.32545	0.34368	0.35627
2	0.2731	0.30065	0.32495	0.34845	0.36175

Table 3.5: The validation error of the 5 best regions for different C_{back}

Figure 3.17 plots the results of multiple kernel classification error rates against different levels of C_{back}. It shows that as C_{back} increases, the single kernel

classifier decreases from around 0.39 to 0.36 while the multiple kernel learning errors increase from 0.21 to 0.27 for N larger than 30. When C_{back} is small, little spatial correlation in the background image, the multiple kernel learning outperforms the single kernel learning by reducing 45% of the classification rate. As C_{back} increases, the differences between the multiple kernel learning and single kernel learning become smaller.

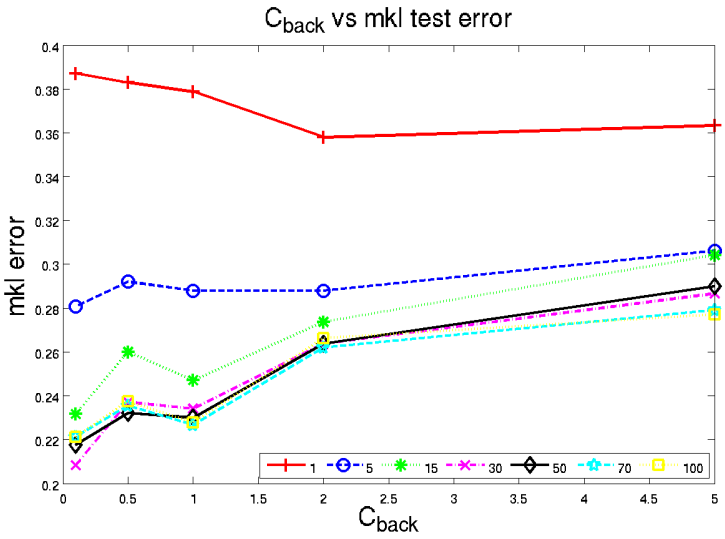

Figure 3.17: Multiple kernel learning results for different C_{back}. Different lines in the figures represent different numbers of top regions used in the multiple kernel learning. In the legend from the top to the bottom, the red line, blue line, green line, magenta line, black line, cyan line, and yellow line correspond to $N = 1$, 5, 15, 30, 50, 70 and 100, separately.

Figure 3.18 compares the weight maps for different levels of C_{back} for $N = 100$. When $C_{back} = 0.01$, the multiple kernel weights are all on the informative regions. As C_{back} increases, the weights are more scattered among both the informative regions and the noninformative regions. This result is consistent with the classification error results in Figure 3.17. As C_{back} increases, it is harder to distinguish the informative regions from the noninformative regions.

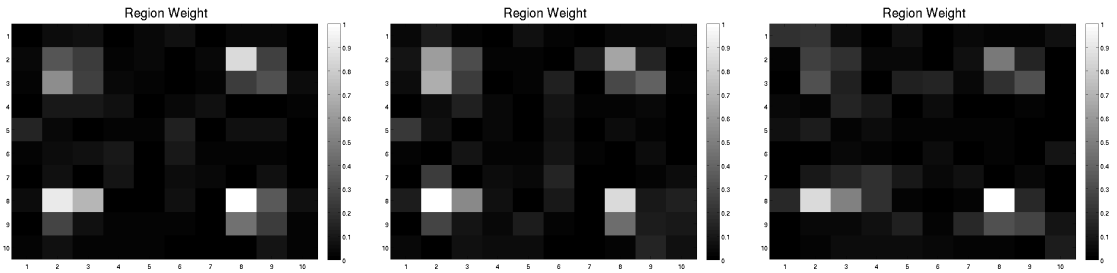

Figure 3.18: Region weight map for different C_{back}. From the left to right, $C_{back} = 0.01$, 2 and 10 and all the weights are scaled to be in $[0, 1]$.

5. C_{inf}

   C_{inf} is the parameter controlling the spatial correlation in the informative images. The larger the C_{inf} gets, the stronger the spatial correlation in the informative regions is. Figure 3.19 shows the informative regions of both the control and patient groups for different values of C_{inf}. When $C_{inf} = 0.1$ the informative regions are more like independent values while the informative regions in the last column are more like constants. All other parameters used to generate the data are fixed $\sigma_{noise} = 0.7$, $\mu = 0.05$, $\sigma_{inf} = 0.05$ and $C_{back} = 2$.

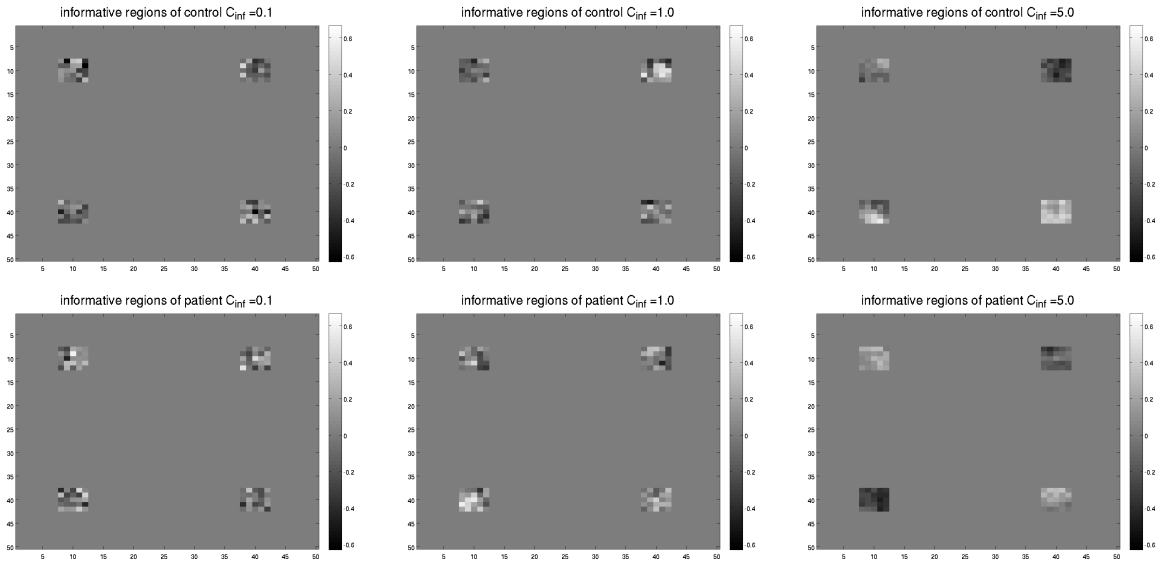

Figure 3.19: Images of the informative regions for different C_{inf}. From left to right C_{inf} are 0.1, 1, and 5. The upper row are the images of control group and the lower row are the images of patient group.

Table 3.6 shows the validation error rate of the top 5 regions for different values of C_{inf}. The individual error rate decreases as C_{inf} increases. When $C_{inf} = 0.1$, there exists little correlation in the informative region. The best error rate is around 0.3. As C_{inf} increases to 5, the error rate goes down to around 0.21.

C_{inf}	1	2	3	4	5
0.1	0.30427	0.31528	0.33615	0.3502	0.36245
0.5	0.29708	0.31465	0.33422	0.3426	0.35562
1	0.27237	0.30173	0.32223	0.34443	0.35665
2	0.23515	0.28643	0.3154	0.33307	0.35442
5	0.21322	0.25653	0.29278	0.33255	0.34942

Table 3.6: The validation error of the 5 best regions for different C_{inf}

Figure 3.20 plots the results of multiple kernel classification against different levels of C_{inf}. As C_{inf} increases, the multiple kernel error decreases. For $C_{back} = 0.1$, the multiple kernel learning test error rate is 0.43 for $N = 1$ and 0.27 for $N \geq 30$. As C_{back} increases to 5, the test error rate is 0.26 for single region and around 0.2 for more than 30 regions.

Figure 3.21 compares the weight maps for different levels of C_{inf} for $N = 100$. As C_{inf} increases, the weights are more concentrated on the informative regions which reduces the multiple kernel error rate. A high value of C_{back} creates more correlation in the informative regions and enhances the differences between the two groups. So a large C_{back} makes the informative regions more distinguishable from noninformative regions.

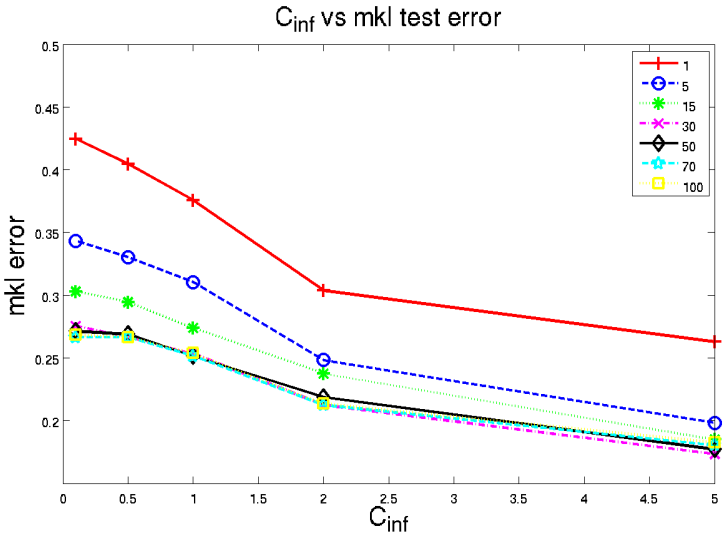

Figure 3.20: Multiple kernel learning results for different C_{inf}. Different lines in the figures represent different numbers of top regions used in the multiple kernel learning. In the legend from the top to the bottom, the red line, blue line, green line, magenta line, black line, cyan line, and yellow line correspond to $N = 1$, 5, 15, 30, 50, 70 and 100, separately.

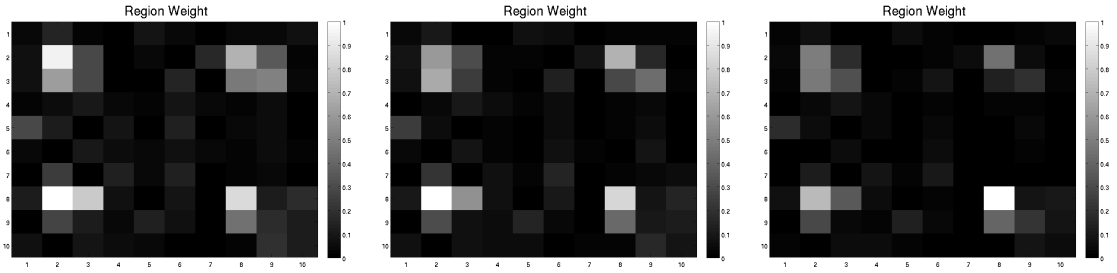

Figure 3.21: Region weight map for different C_{inf}. From left to right $C_{inf} = 0.1$, 1 and 5.

3.5 Real Data Analysis

In this section, we modified our two-step procedure proposed in section 3.4.2 and tested it on four brain image data sets. The goal was to to classify the patients and healthy controls using only the whole-brain anatomical magnetic resonance images. We started from building linear kernel on small regions in the brain and then combining them in the multiple kernel learning step. Section 3.5.1 introduced four data sets of different diseases and the preprocessing steps on the scans. Section 3.5.2 explained the two-step procedure in details, including brain parcellation, training and test split, first-step single-kernel SVM and the multiple kernel step. Section 3.5.3 presented the results of the two-step procedure on different levels. We compared the results between the linear kernel and Gaussian kernel and tried different dimension reduction methods to achieve a better classifier. In addition to the performance of the classifiers, we also presented the significant regions using different kernels and features.

3.5.1 Data and Preprocessing

In this work, we used four data sets to test the multiple kernel classifier on different kernels and features. The data sets were Alzheimer's disease, mild cognitive impairment, Systemic Lupus Erythematosus disease and chronic pelvic pain disease.

Alzheimer's disease and Mild Cognitive Impairment Alzheimer's disease (AD) is the most common form of dementia and the sixth-leading cause of death in the United States. It is the leading cause of dementia which usually causes symptoms such as confusion, irritability and aggression, mood swings, trouble with language, and long-term memory loss. Mild cognitive impairment associated with an increased risk of conversion to AD (*Petersen et al.*, 1997) is considered as a prodromal state of AD (*Schroeter et al.*, 2009). Recently, there are lots of studies showing that besides neuropsychological examination, structural images of the brain can support the diag-

nosis of the AD and MCI. We tried to follow this approach, designing a classifier that could help in the clinical diagnosis. The data used here were from the Alzheimer's disease Neuroimaging Initiative (ADNI) database (*Mueller et al.*, 2005) which provided a generally accessible data repository to assist the research of Alzheimer's disease and MCI disease. We obtained 58 healthy subjects, 80 MCI patients and 36 Alzheimer's patients from ADNI.

Systemic Lupus Erythematosus Systemic lupus erythematosus disease (SLE) is an autoimmune disease which presents a wide range of symptoms, such as fever, malaise, joint pains, myalgias, fatigue, and temporary loss of cognitive abilities. Since these symptoms are so often seen with other diseases, SLE still presents very difficult diagnostic challenges. It is reported that subtle changes in regional brain structure are often observed for SLE patients. Brain image, especially MRI, is frequently used as a routine investigation. Here, we collected 18 SLE patients along with 19 healthy controls which were matched to the patients in terms of age, gender and education levels. None of the patients had Neuropsychiatric systemic lupus erythematosus and No cerebral atrophy was found in patient brain. A visual assessment of the brain didn't reveal any significant differences between two groups. A detailed description of the data can be found in Cagnoli's paper (*Cagnoli et al.*, 2012). We applied machine learning method on SLE data to see if the quantitative method could discover anything that visual examination failed to find out.

Chronic Pelvic Pain Chronic pelvic pain (CPP) is defined as "non-cyclic pain of 6 or more months" duration that localizes to the anatomic pelvis, anterior abdominal wall at or below the umbilicus, the lumbosacral back, or the buttocks, and is of sufficient severity to cause functional disability or lead to medical care (*ACOG Committee on Practice Bulletins–Gynecology.*, 2004). As in other chronic pain diseases, CPP is not only associated with the presence of peripheral pathology but also related to the

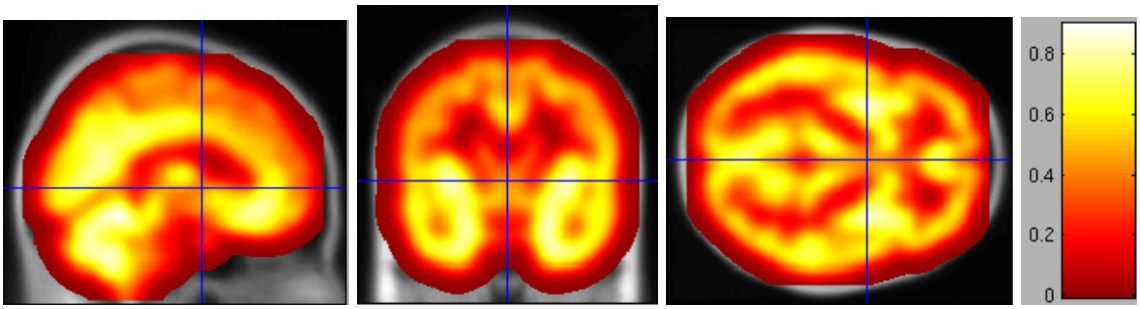

Figure 3.22: Preprocessed image

central nervous system which may amplify the pain processing. Therefore, MRI scans might help in understanding the pathogenesis of CPP (*As-Sanie et al.*, 2012). We examined changes in the brain image for CPP patients to build the connection between CPP and pain matrix. The data used here is from department of Obstetrics and Gynecology, University of Michigan. A detailed description of the data can be found in (*As-Sanie et al.*, 2012)In this study, we had 17 women with endometriosis-associated CPP and 25 healthy controls.

Preprocessing The data used were all T2-weighted images of the whole brain. Only the first images were used if there were multiple images of the same subject acquired at different times. The preprocessing of the data involved both linear and nonlinear registration, mapping the images to the stereotactic space defined by the Montreal Neurological Institute (MNI; www.loni.ucla.edu/ICBM/ICBM_Databases.html). After registration step, all images were smoothed by convolving with an isotropic Gaussian kernel of 10 mm full-width at half maximum (FWHM). Figure 3.22 shows a preprocessed scan of a healthy subject.

3.5.2 Methods and Algorithm

For anatomical image, a single scan usually contains hundreds of millions of voxels. The amount of noise voxels is much larger than the informative voxels. Traditional SVM classifier does not perform well since it treats all voxels as a long vector and the

noise voxels will mask the informative voxels. One solution to this is using features particular to the problem, such as using voxels in a particular region of interest (ROI) other than the whole brain or using some model statistics other than the original voxel values. But this feature selection prior to applying a classification method needs expertise information about the underlying problem and may be subject to selective bias. However if we don't have this extra information, we need a systematic and automatic way to select the important features. Since the multiple kernel learning can distinguish the important kernels from the uninformative one through learning the kernel weights η_m, we can design different kernels on different local regions to select the significant regions. In addition to a better classifier with high accuracy, we also can identity the significant regions corresponding to the disease through the multiple kernel learning method.

In this study, we only used the voxels in the gray matter which were defined as any voxels with values above 0.1. The ideal situation is to design one kernel on all voxels that contain similar information about the class label and have different kernels on regions that demonstrate different aspects of the data structure. The localized individual kernels gather the information of several similar voxels to enhance the signal to noise ratio and the multiple kernel learning combines different local information together to get a better classifier. In reality, it is hard to segment the regions into local functional regions. As voxels physically close are tend to belong to the same functional regions, we segmented the gray matter of the whole brain into smaller non-overlapped regions and built a linear kernel on each region. This segmentation greatly reduced the dimension of the original problem. Since most regions do not contain valuable information and the multiple kernel learning can not handle large amount of kernels, we used individual SVM to select the significant regions for multiple kernel learning analysis.

We now described the two-step procedure that was applied to the data. We

were given a classification problem with N_p patients and N_c healthy controls. For evaluation purpose, 15% subjects from each group were left out from each group to test the final multiple kernel classifier. The remaining data were divided into individual validation set with 15% subjects from each group and individual training set with 70% subjects from each group. At the first step, a traditional SVM was applied to each individual region, trained on the individual training data and testing on the validation data. This gives us an accuracy map of the whole brain. Then M significant regions with the lowest error rates were chosen based on their individual error rates to further break down into smaller cubes of $5 \times 5 \times 5$. And then a single SVM was applied to each cube to get a validation error rate for the each cube in those M regions. At the second step, a multiple kernel learning was applied to the top K cubes with the lowest error rate, training on the training data and then testing on the test data set. The test and training split were repeated 300 times to produce a mean classification error rate of the two-step procedure. In addition to the classifier, we also had a weight map of the top K cubes in each split. We averaged over 300 splits to get a weight map of the cubes with lowest individual error rates. This procedure is explained in figure 3.23.

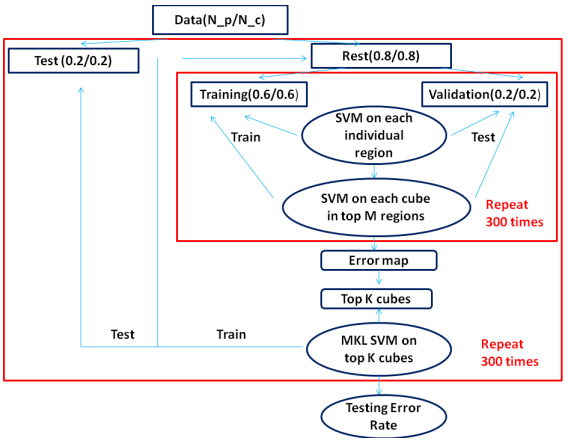

Figure 3.23: The float chart of the method. The data are divided into testing set (15%), validation set (15%) and training set (70%). At the first step, a single kernel SVM is applied to each individual region, trained on the training set and tested on the validation set. The training and test splits are repeated 50 times to get an average of validation error rate for each region. Then the top M regions with the lowest error rates are divided into $5 \times 5 \times 5$ cubes. A single kernel SVM procedure is again trained on each cube to get a validation error rate for all the cubes in the top regions. At the second step, a multiple kernel SVM is trained on the top K cubes in those M regions to get a final classifier. The multiple kernel error rate is found by training on the training and validation sets together and testing on the test sets.

3.5.3 Results

In this section, we presents the results of the two-step procedure to the four data sets. We examined the performance of multiple kernel classifier mostly on two aspects, the misclassification error rate and the identified significant regions. We further improve the classifier by considering different challenges of the problems, such as feature selection, kernel selection and tuning parameter selection.

Regions

At the first step, each scan was first divided into functional regions according to MNI brain atlas (*Tzourio-Mazoyer et al.*, 2002). Then a single linear kernel SVM was trained on each regions. The result is a region error map with an error rate for each region. Figure 3.24 shows the region error map for SLE data.

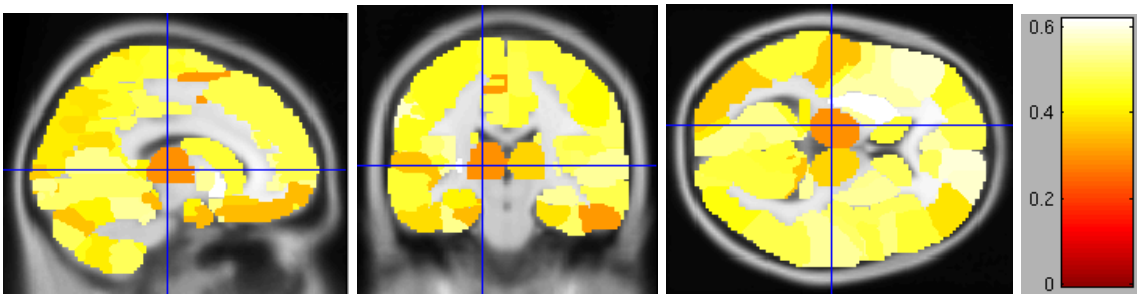

Figure 3.24: Individual region error map of SLE data. The three figures are 2-dimensional cross-sections through a voxel in the brain labeled by a cross in the figures. From left to right are the sagittal view, the coronal view and the axial view separately. The color represents the misclassification error rate on the validation set. The region with the darkest color and point by the cross is the left thalamus.

From Figure 3.24, we can see most regions have classification error rates around 0.5. Only a few regions show error rates significant lower than random guess. Left thalamus shows a lower error rate of 0.2851. It is observed that significantly smaller thalamic volumes happened in SLE patients when compared to healthy controls (*Appenzeller et al.*, 2009). Left supplementary motor area also has a lower error rate

of 0.2973 and it is shown to be related to the pain caused by the SLE disease. Parahippocampal and precuneus regions also show lower-than-average classification error rates comparing to other regions in the brain. Both regions have been reported in the diagnosis of SLE literature. Parahippocampal regions are found significant in SLE in the work by (*Cagnoli et al.*, 2012) and precuneus regions are related to the memory impairment of the SLE patients (*Oh et al.*, 2011).

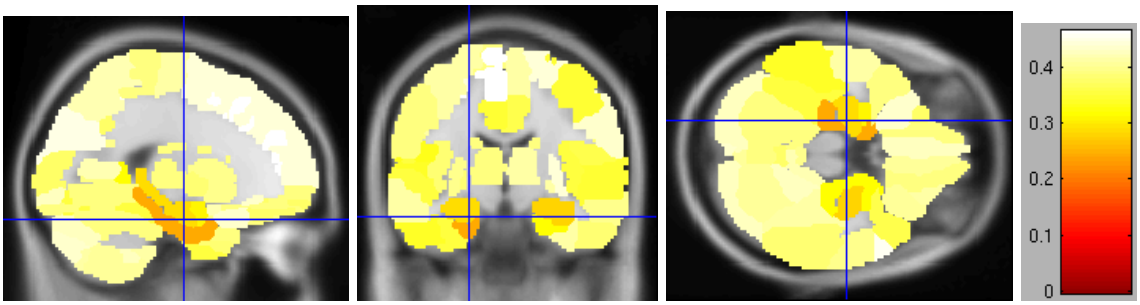

Figure 3.25: Individual region error map of AD data. The three figures are 2-dimensional cross-sections through a voxel in the brain labeled by a cross in the figures. From left to right are the sagittal view, the coronal view and the axial view separately. The color represents the misclassification error rate on the validation set. The region labeled by the cross in the figures is the hippocampal regions. The region with the darkest color under hippocampal in the first figure is the parahippocampal region.

Figure 3.25 applied the same procedure to the AD data, showing the importance of each region. For AD, the regions achieving the lowest error rates are left and right parahippocampal with error rates of 0.2381 and 0.2752. Several studies report the difference in parahippocampal regions between two groups (*Van Hoesen et al.*, 2000). And the left and right hippocampal regions also have error rates around 0.3. Change in hippocampal's volume is also reported to be a sensitive marker for pathological AD stage (*Gosche et al.*, 2002).

Figure 3.26 shows the region error map for MCI. For MCI data, some significant regions are the same as in the AD case, such as parahippocampal regions and hippocampal regions. But the regions' individual error rates are higher than the ones in AD, with error rates around 0.35. The right amygdala region shows a lower-than-

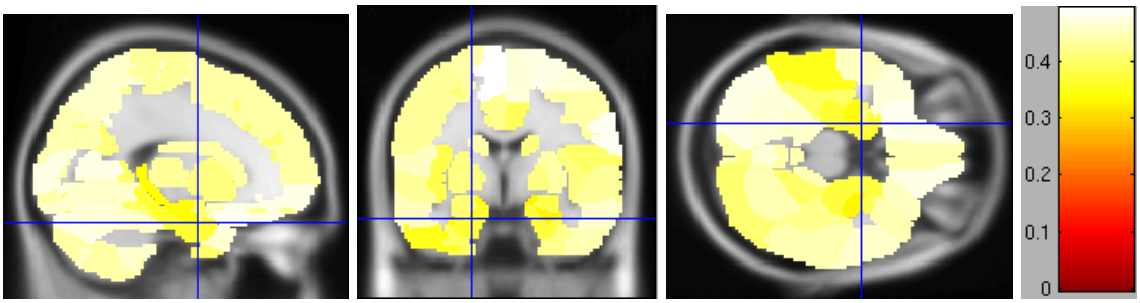

Figure 3.26: Individual region error map of MCI data. The three figures are 2-dimensional cross-sections through a voxel in the brain labeled by a cross in the figures. From left to right are the sagittal view, the coronal view and the axial view separately. The color represents the misclassification error rate on the validation set. The region labeled by the cross is the parahippocampal region.

average classification rate which was observed in other study (*Pennanen et al.*, 2005).

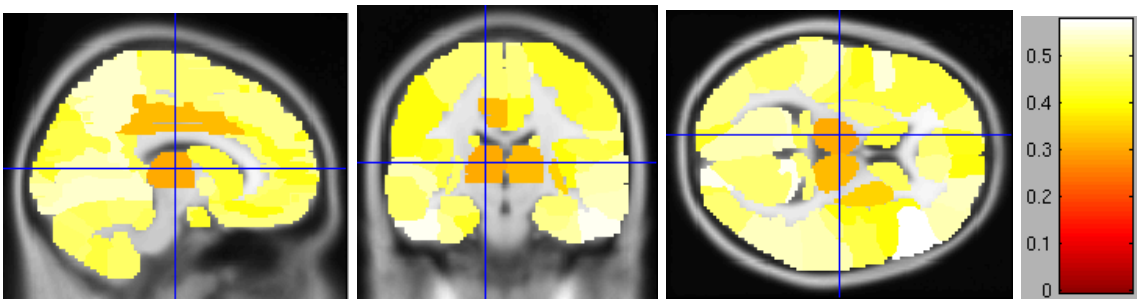

Figure 3.27: Individual region error rate map of CPP. The three figures are 2-dimensional cross-sections through a voxel in the brain labeled by a cross in the figures. From left to right are the sagittal view, the coronal view and the axial view separately. The color represents the misclassification error rate on the validation set. The region labeled by the cross is the left thalamus.

Figure 3.27 shows the regions error rate map for CPP using linear kernel SVM. For CPP, the region achieving the lowest error rate is cerebellum region with an error rate of 0.2808. There are several works suggesting the connection between cerebellum and pain perception (*Moulton et al.*, 2010). Both left and right thalamus regions have error rates around 0.3. It is long known that the thalamus regions are part of the pain matrix. These regions are also found to be related to CPP in other works (*As-Sanie et al.*, 2012).

Cubes

The region unit in the above analysis is usually a complicated structure with many small functional areas. In this step, we segmented the regions further into smaller cubes to identify finer areas that were associated with the disease. For each training and validation split, we first trained an individual SVM on each region and then chose the top K regions with the lowest individual classification errors. Each region was then segmented into $5 * 5 * 5$ mm non-overlapped cubes as described in section 3.5.2 and a single kernel SVM was built on each cube to get an error rate for each cube. All SVM classifiers were trained on the training set and tested on the validation set. The result was an error map of individual error rates for cubes in the significant regions. Figure 3.28 showed the individual cube error maps for four data sets. From top to bottom are the maps of SLE, AD, MCI and CPP data sets. From those figures we can tell that the top cubes in the significant regions are clustered together rather than scattered across the whole brain. This shows that the region error rates are stable across different training and validation splits. The regions with lower classification error rates in one training and validation split are more likely to be in the top K in another split. So these cubes are picked for individual SVM more frequently than others.

From the cube error maps, we can see that not all the voxels in the cube have similar classification power. For SLE data, thalamus shows up as a significant region in different training and validation splits. Most of the cubes in the thalamus are picked as significant cubes but their average classification error rates range from 0.25 to 0.4. This means not all the parts in thalamus play the same role. In order to identify significant areas with better spatial resolution, it is better to use a finer unit than the regions defined by MNI brain atlas. For thalamus in SLE data, the cube achieving lower classification error is in pulvinar nuclei in thalamus. And for AD and MCI patients, the most significant cubes are in parahippocampal gyrus in

parahippocampal region. For CPP, the cubes with the lowest classification error rates are in putamen and thalamus.

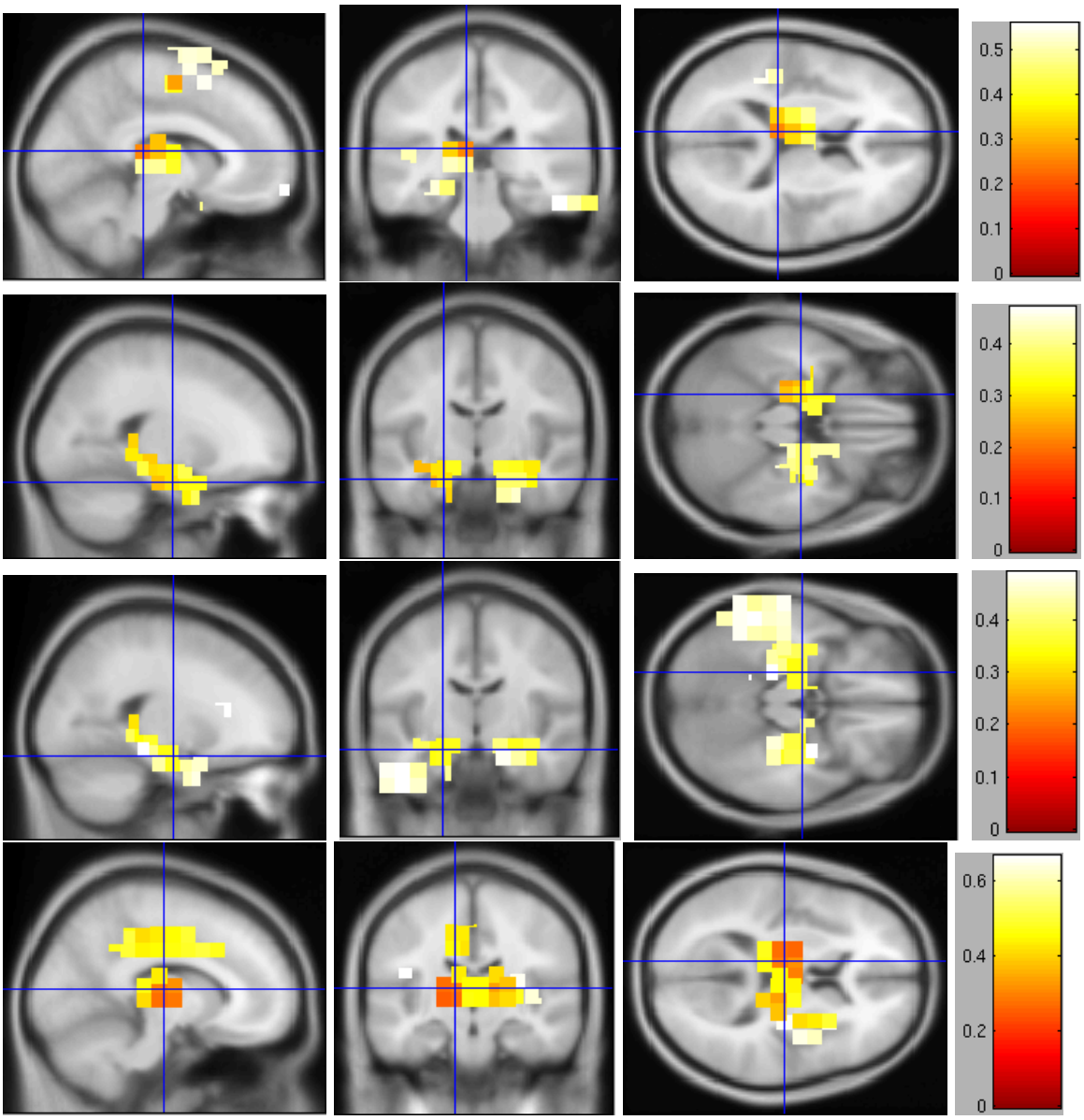

Figure 3.28: Cube error maps of the top cubes for four data sets. From top to bottom are the maps of SLE, AD, MCI and CPP data sets. The three figures are 2-dimensional cross-sections through a voxel in the brain labeled by a cross in the figures. From left to right are the sagittal view, the coronal view and the axial view separately. The figures show the top 100 cubes with the lowest validation error rates in the top regions.

Classification Error Rate

After the individual cube error maps, we selected the top K cubes with the lowest classification error rates to build the MKL SVM. Let $\{\overrightarrow{\mathbf{x}_{n1}}, \overrightarrow{\mathbf{x}_{n2}}, \ldots, \overrightarrow{\mathbf{x}_{nM}}\}$ be the set of vectors on the top K cubes for subject n. $\overrightarrow{\mathbf{x}_{nk}}$ represents the vector of voxels in the k-th cube of subject n. For MKL SVM, we built a single kernel on each cubes. The linear kernel for cube k between subject i and j was defined as:

$$H_k(i,j) =< \overrightarrow{\mathbf{x}_{ik}}, \overrightarrow{\mathbf{x}_{jk}} >$$

Then the final multiple kernel $H(i,j) = \sum_{k=1}^{K} \eta_k H_k(i,j)$ found the best linear combination of K kernels to maximize the optimization function (3.3). The multiple kernel classifier was trained on the training and validation set together and tested on the test set. The following table reports the classification error rate for four data sets under different values of M and K. Tuning parameter C for SVM was 10.

Table 3.7 compares the results of traditional SVM method to the two-step procedure. The first row in the Table 3.7 are the error rates of using single kernel SVM on the whole brain. The rest of the results are the multiple kernel learning error rates from the two-step procedure with $M = 5$. The results show that in all the four cases, two-step procedure achieve an error rate lower than single kernel SVM. This confirms

C=10		SLE	AD	MCI	CPP
SVM		0.516	0.4002	0.4311	0.4444
M = 5	K = 10	0.3995	0.3158	0.3659	0.3735
M = 5	K= 20	0.3935	0.3118	0.3700	0.3737
M = 5	K = 50	0.451	0.3729	0.4293	0.4944
M = 5	K = 100	0.423	0.3676	0.4322	0.4897

Table 3.7: MKL error rates of four data sets, $M = 5$. The first row is the result of the traditional SVM, treating all the voxels as a long vector. The number of top regions to get the small cubes is $M = 5$. The number of top cubes in the final classifier is $K = 10$, 20, 50 and 100. The red color in each column highlights the lowest classification error rate for each data.

that the multiple kernel learning SVM can get a better performance by taking the structural information into consideration.

Moreover, Table 3.7 shows that brain images have different classification power for different diseases. AD is the easiest to classify with the lowest classification error rate since it is a neurodegenerative disease which directly relates to the loss of the gray matter in the brain. MCI is similar to AD but with weaker symptoms and signals. So it is more difficult than AD. SLE has the highest classification error rate because for the SLE data we had very few subjects to train the classifier. So it is very hard to identify the true signal from the noise for such a high-dimension problem. Another observation is that for a small number of regions ($M = 5$), adding more cubes in the final multiple kernel analysis actually leads to a higher error rate. This means different parts in the region are not the same and selecting the insignificant cubes in the significant regions won't improve the classifier.

We then compare the cases of $M = 10$ and $K = 10$, 20, 50, 100 to the classifier of a single cube. Table 3.8 shows the MKL error rate for $M = 10$ for four data sets. The first row is the error rate corresponding to $M = 1$ and $K = 1$ which means at each test split, only the cube with the lowest validation error rate is used for the classifier. Multiple kernel learning achieves a lower classification error than single kernel classifier for all the four data sets and all the values of M. This shows combining informative cubes together can give a more informative classifier.

C=10		SLE	AD	MCI	CPP
M = 1	K = 1	0.4138	0.3577	0.3982	0.4522
M = 10	K = 10	0.3885	0.318	0.3664	0.3633
M = 10	K = 20	0.3875	0.3150	0.3779	0.3622
M = 10	K = 50	0.3575	0.3226	0.3797	0.3764
M = 10	K = 100	0.3482	0.3325	0.3722	0.3811

Table 3.8: MKL error rates of four data sets, $M = 10$. The number of top regions to get the small cubes is $M = 10$. The number of top cubes in the final classifier is $K = 10$, 20, 50 and 100. The red color in each column highlights the lowest classification error rate for each data. The first row is the test error for single kernel SVM on the top cube with the lowest validation error.

Dimension Reduction

Selecting top cubes from all the regions in the brain didn't give a good classifier. This was because of the high dimensionality of the brain image data and the small sample size of the problem. The signals were weak comparing to noise and only in few voxels. So it was very hard to pick up the signals from the whole brain using only few training subjects. In order to solve this problem, we tried two different approaches here to reduce the dimensionality of the problem. One was to do feature selection. Instead of using the original signals, we used a vector as a very coarse estimation of the density function of each cube. The kernel took the estimates of the density function not the voxels values. So this method extracted new features from the original data to reduce the dimension. Another solution was to start from a few informative regions rather than all the regions in the brain. This method reduced the dimension by borrowing extra information from expert knowledge. In this study, we chose six informative regions for different disease according to the literature about the disease.

Feature Selection For the feature selection method, we used an equally spaced 10-bin histogram to represent each cube. The features for the cube k was a vector

$\overrightarrow{z} = \{z_1, z_2, \ldots, z_{10}\}$, with $\sum_{i=1}^{10} z_i = 1$. Since all the values were normalized between 0 and 1, z_i was the proportion of voxels with signal values between $0.1 * (i-1)$ and $0.1 * i$. Then the kernels were built on $\overrightarrow{z}$ rather than original signal $\overrightarrow{x}$.

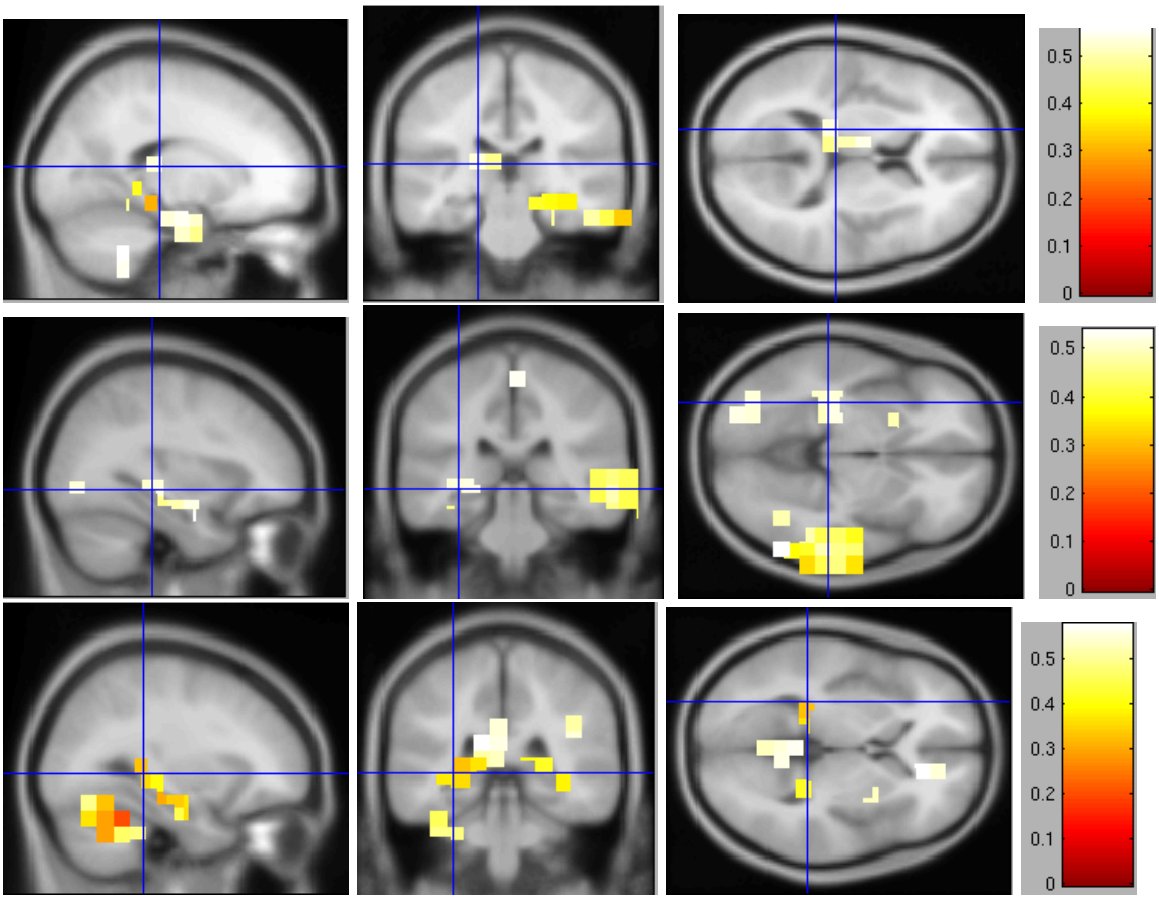

Figure 3.29: Cube error maps of three data sets, using density feature. From the top to the bottom are SLE, AD and MCI separately. First, each region is represented by a 10-bin histogram of all its voxels and then a single kernel SVM is trained on each region. Top M regions are selected to further break down into small $5 \times 5 \times 5$ cubes. Then each cube is again represented by a 10-bin histogram of its voxels and an SVM is trained on each cube. The figure presents the classification error rate of the top 100 cubes with the lowest error rates.

We repeated the two-step procedure taking density estimates as features instead of original signal to produce another cube error map. Figure 3.29 shows the cube error maps for the three data sets using density features. From the top to the bottom are SLE, MCI and AD datasets. Comparing Figure 3.29 to 3.28, we can see there

are some cubes showing significant in both Figures. Figure 3.29 also confirms the thalamus regions for SLE data and parahippocampal regions for AD and MCI data sets. It also reports some regions that are not identified by the classifier taking original signals, such as the temporal regions in the SLE data. The selected cubes are more scattered across the whole brain rather than concentrated in few regions as Figure 3.28.

After getting the top cubes, we trained a MKL on the top cubes, taking only the density estimates. Table 3.9 shows the multiple kernel classification error for three data sets using density as features. Comparing to Table 3.7, density features doesn't achieve a better classifier. For SLE data, the best error rate is around 0.38 while using the original signal the error rate is around 0.34. For AD data, the best error rate is around 0.35 for density features and the best error rate is around 0.31 for signal features. For MCI, the error rate is higher than AD data, with a value around 0.38 for density features and a value around 0.36 for using original signal. This may suggest that a 10-bin vector can not represent the density function of each kernel well. AD is still the one with the lowest classification error rates and SLE data was the one with the highest error rates.

		Density Feature			Original Signal		
C=10		SLE	AD	MCI	SLE	AD	MCI
M=10	K=10	0.4455	0.3605	0.4038	0.3885	0.318	0.3664
M=10	K=20	0.4317	0.3658	0.3936	0.3875	0.3150	0.3779
M=10	K=50	0.4307	0.3599	0.3831	0.3575	0.3226	0.3797
M=10	K=100	0.411	0.3692	0.3787	0.3482	0.3325	0.3722

Table 3.9: Multiple kernel classification error rates of three data sets, using density features. Each region was first represented by a 10-bin histogram and then trained an SVM classifier to get a validation error rate map of the regions. The top regions with the lowest error rates were broken down into small cubes. Then each cube was represented by a 10-bin histogram to train a single kernel SVM to get validation error map for the cubes. The MKL SVM takes the 10-bins histogram of the top cubes to train a classifier. The tables shows the classification error rate for both density features and original signals. The red color highlights the lowest error rate of each column.

Region Selection In this part, we focused our analysis on a few regions instead of the whole brain. We choose six informative regions for each disease according to the extra knowledge about the disease. The regions selected for SLE is left and right thalamus ares, left and right supplementary motor areas and left and right precuneus areas. The regions selected for AD is left and right parahippocampal areas, left and right lingual areas and left and right precuneus areas. The regions selected for MCI are left and right parahippocampal areas, left and right hippocampus areas and left and right precuneus areas. The regions are selected before the analysis. We first applied a single kernel SVM on each of the six informative regions to get a classification error for each region. Figure 3.30 shows the region error rate map for informative individual region using linear kernel. From Figure 3.30, we can see not all the pre-selected regions have small classification error rates. The thalamus regions in the SLE and the parahippocampal regions in AD and MCI have relative small classification error rates.

We then selected the top three regions with the lowest classification error rates for each training and test split and divided the selected region into small $5 * 5 * 5$ cubes.

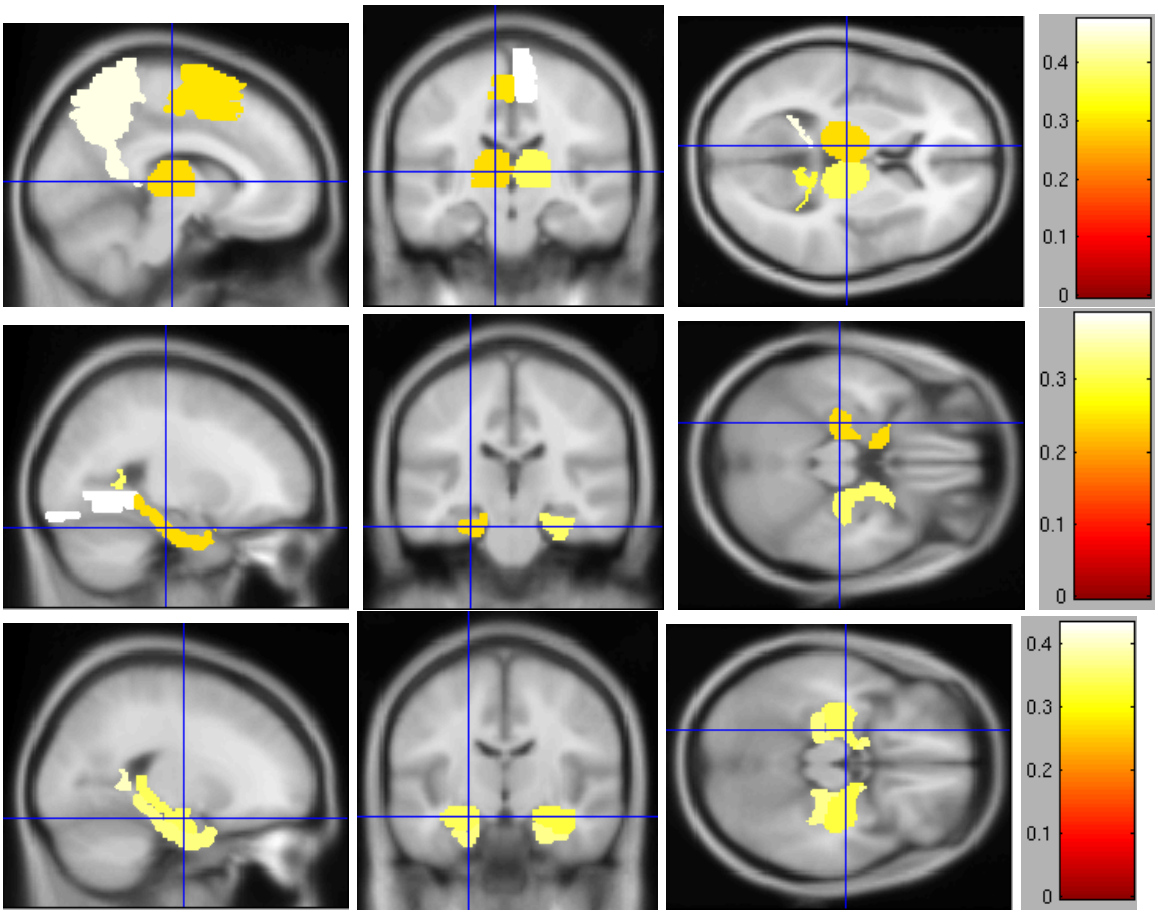

Figure 3.30: Region error maps of six informative regions. A single SVM is applied to each of the six informative regions to get a validation error rate for each region. The color shows the validation error rates for those informative regions. From the top to the bottom are SLE, AD and MCI data sets. The SLE figures show the left thalamus under the cross. The AD figures show the left hippocampus area under the cross. The MCI figures show the left hippocampal area under the cross.

And then we applied SVM again on the cubes to get the top cubes. Figure 3.31 shows the cube error rate map for informative individual regions using linear kernel. From the top to the bottom are the cube error rate maps of SLE, AD and MCI.

From Figure 3.31, we can see the cubes have different classification error rates. Comparing Figure 3.31 to Figure 3.28, the significant cubes tend to be stable across different splits. For AD and MCI, the cubes in the parahippocampal gyrus have small classification error rates in both figures.

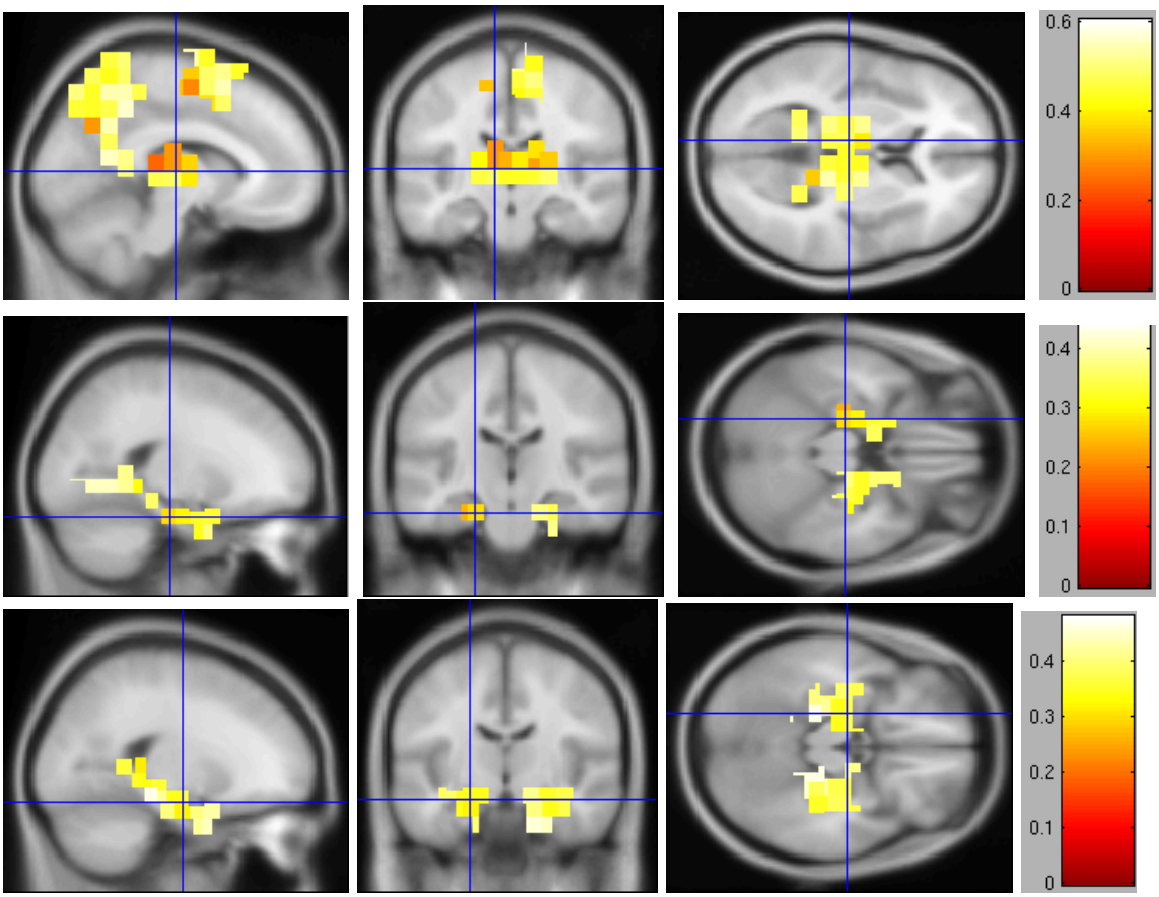

Figure 3.31: Cube error maps of cubes in informative regions. A single SVM is applied to all the cubes in the top three regions with the lowest validation error rates. The figure shows the validation error rates for those cubes. From the top to the bottom are SLE, AD and MCI data sets.

After getting the cube error rate maps, we applied a multiple kernel analysis on the top cubes. Table 3.10 shows the multiple kernel error rates for three data sets. We compare the classification error rate of applying the two-step procedure on only the informative regions to the method starting from all the regions. Table 3.10 shows that using a few selected information regions can improve the performance of the classifier. The classification error rate is around 0.37 for SLE data when using all the regions in the brain scan with $M = 20$ and $K = 50$. But the rate is only 0.32 when using only the informative regions for $K = 50$. And for AD, the error rates are around 0.31 when using all the regions and decrease to 0.30 for $K = 20$. For MCI,

	$C = 10$	Informative	All Regions
	M	3	20
SLE	K = 20	0.32375	0.3932
	K = 50	0.31925	0.3762
AD	K = 20	0.2991	0.3132
	K = 50	0.3234	0.3116
MCI	K = 20	0.34007	0.3849
	K = 50	0.34436	0.3840

Table 3.10: MKL error rates of on informative regions. We applied the two-step procedure on six informative regions that are selected according to the extra knowledge about the disease. At the first step, A single kernel SVM is applied to the regions and then three regions with the lowest error rates are broken down into smaller cubes. Then a single kernel SVM is applied to all the cubes in the selected regions. The final classifier is built on K cubes with the lowest error rates. We compare the method of using only the informative regions with the method starting from the whole brain.

the previous error rates were around 0.38 for all the regions and were around 0.34 for informative regions. This shows that focusing on the informative regions can reduce the dimension, making the classification an easier task.

Kernel Selection

For the same features, we can also design different kernels to find the best classifier. Linear kernel is the simplest one of which the feature space is just the original signal. And the final classifier is a hyperplane in the original spaces. In this study, we also tried the Gaussian kernel which produced more complex features than the linear kernel and could fit any boundaries. The Gaussian kernel taking two vectors $\overrightarrow{\mathbf{x}_i}$ and $\overrightarrow{\mathbf{x}_j}$ is defined as:

$$H(\overrightarrow{\mathbf{x}_i}, \overrightarrow{\mathbf{x}_j}) = \exp\{\frac{||\overrightarrow{\mathbf{x}_i} - \overrightarrow{\mathbf{x}_i}||^2}{C_{gauss}}\}$$

The complexity of the kernel is controlled by parameter C_{gauss}. The smaller the C_{gauss} gets, the more complex the classifier is. We used cross-validation to find the tuning parameter C_{gauss} for C_{gauss} = 1, 10 and 50. We applied the same procedure to the

four data sets using a Gaussian kernel. We reported the region error maps, the cube error maps and the multiple kernel learning error rates.

Region Error Map We applied the two-step procedure to the brain with Gaussian kernel. At the first step, a single kernel SVM with Gaussian kernel was applied to each region in the whole brain

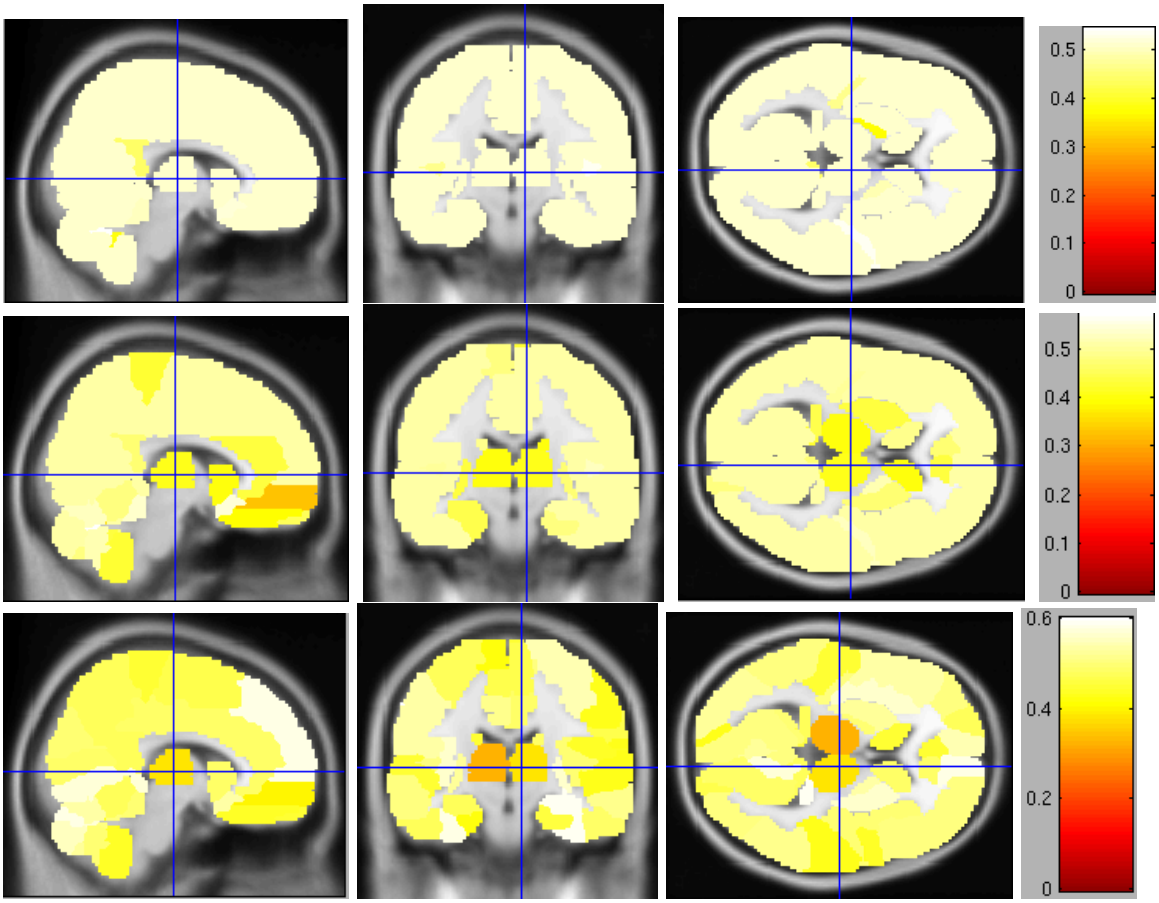

Figure 3.32: Region error maps for SLE data, using Gaussian kernel. Color represents the misclassification error. Figures in each row represent 2-dimensional cross-sections through the point labeled by a cross. From the left to right are the sagittal view, the coronal view and the axial view. From the top to the bottom are the figures corresponding to $C_{gauss} = 1$, 10 and 50.

Figure 3.32 shows the individual error rates for each region for SLE data. The first row is the error rate map for $C_{gauss} = 1$. The second row is the error rate map for $C_{gauss} = 10$ and the last row is the error rate map for $C_{gauss} = 50$. From Figure

3.32, we can see that when $C_{gauss} = 1$, the classifier is too simple to separate the two groups. So all the regions have similar error rate, close to 0.5. As C_{gauss} goes up, the classifier becomes more complex and able to distinguish the patients from the healthy controls. When $C_{gauss} = 50$, the Gaussian kernel can tell the informative regions from the non-informative one. Thalamus regions show a better classification error than other regions in the classifiers using $C_{gauss} = 50$.

Cube Error Map After getting the top regions, we also tested the Gaussian kernel on individual cubes. For each split, we selected the top M regions and divided each region into $5 * 5 * 5$ cubes. A single SVM with Gaussian kernel was applied to each individual cube. And the cube error rate maps plot the top cubes with their error rates.

Figure 3.33 shows the cube error rate maps for SLE data set using Gaussian kernels for different C_{gauss} values. From the top to the bottom are the Gaussian kernels with $C_{gauss} = 1$, $C_{gauss} = 10$ and $C_{gauss} = 50$. All three figures have similar patterns across different values of C_{gauss}. They all identify the cubes in the thalamus and the somatosensory cortex. For larger values of C_{gauss}, the informative cubes are more clustered together.

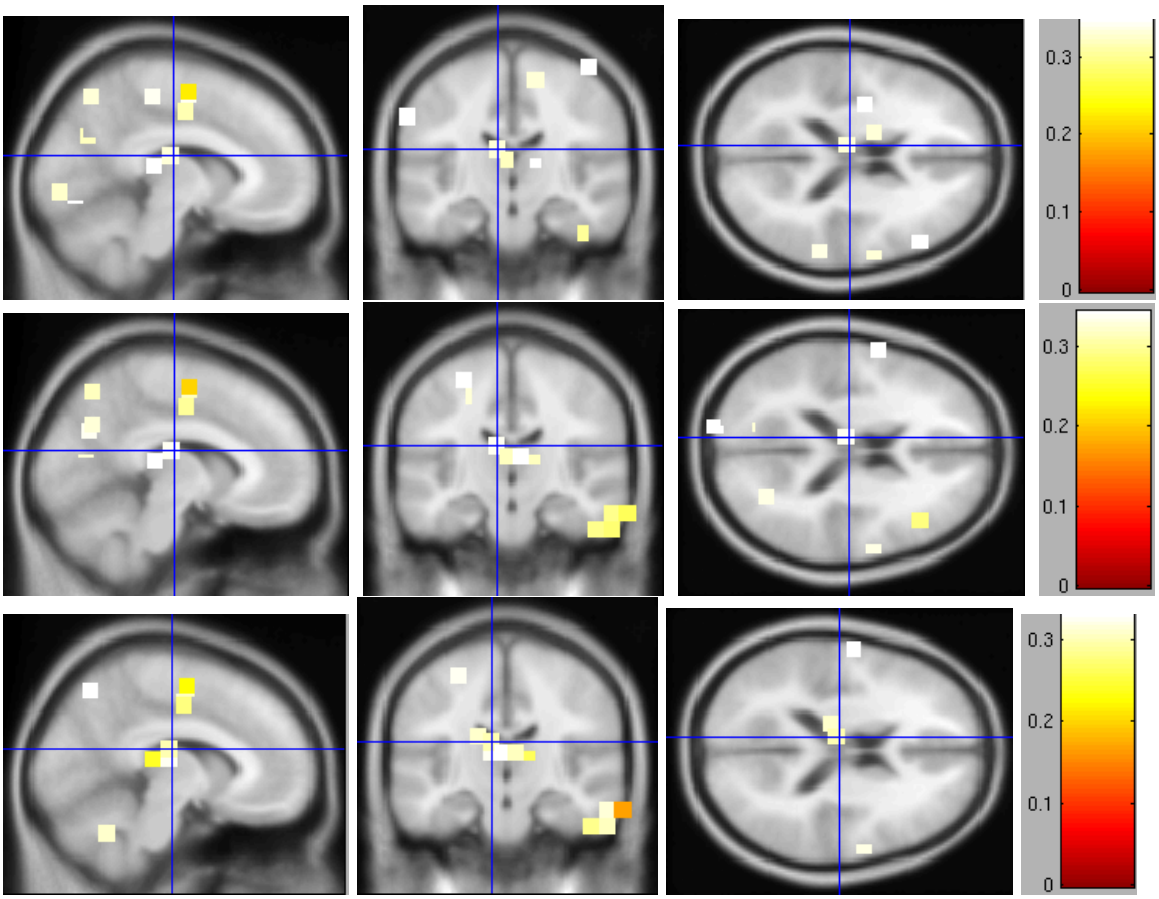

Figure 3.33: Cube error maps for SLE data, using Gaussian kernel. Color represents the misclassification error. Figures in each row represent 2-dimensional cross-sections through the point labeled by a cross. From the left to right are the sagittal view, the coronal view and the axial view. From the top to the bottom are the figures corresponding to $C_{gauss} = 1$, 10 and 50.

Multiple Kernel Error

After getting a cube error rate map, we selected the top regions to build the final classifier. Multiple kernel error rates were found by applying a single Gaussian kernel on top cubes and then combining them in the multiple kernel step. The classifier was trained on the training set and validation set together and tested on the testing set. We compared the multiple kernel error rates of Gaussian kernel with the linear kernel. We presented the results for different tuning parameters C and C_{gauss}.

Table 3.11 shows the multiple kernel learning results for all four data sets, compar-

	$C = 10$		Gaussian			Linear
	C_{gauss}		1	10	50	NA
SLE	M = 5	K = 20	0.476	0.501	0.488	0.393
	M = 5	K = 50	0.459	0.482	0.475	0.451
	M = 10	K = 20	0.485	0.481	0.466	0.387
	M = 10	K = 50	0.468	0.475	0.467	0.357
AD	M = 5	K = 20	0.316	0.359	0.329	0.311
	M = 5	K = 50	0.319	0.336	0.312	0.372
	M = 10	K = 20	0.280	0.330	0.318	0.315
	M = 10	K = 50	0.285	0.325	0.322	0.322
MCI	M = 5	K = 20	0.335	0.367	0.380	0.370
	M = 5	K = 50	0.307	0.353	0.366	0.429
	M = 10	K = 20	0.330	0.365	0.374	0.377
	M = 10	K = 50	0.314	0.362	0.377	0.379
CPP	M = 5	K = 20	0.422	0.477	0.470	0.373
	M = 5	K = 50	0.405	0.477	0.479	0.494
	M = 10	K = 20	0.400	0.48	0.479	0.362
	M = 10	K = 50	0.397	0.488	0.484	0.376

Table 3.11: Multiple kernel learning error rates of four data sets, using Gaussian kernel. We applied the two-step procedure to the whole brain with Gaussian kernel and compare the results to the results of linear kernel. We tried different number of regions and cubes for the classifier, $M = 5$ and 10, $K = 20$ and 50.

ing the results between Gaussian kernel and linear kernel. We tried four combinations, with $M = 5, 10$ and $K = 20, 50$. In SLE and CPP cases, the linear kernel performances better than the Gaussian kernel. The best error rate of SLE was 0.357 for linear kernel with $M = 10$ and $K = 50$. For the same M and K, the best error rate of SLE for Gaussian kernel is 0.467 under $C_{gauss} = 50$. And for CPP, the linear kernel can achieve an error rate of 0.362 with $M = 10$ and $K = 20$. Gaussian kernel gets 0.4 for the same K and M. Since for both SLE and CPP data, we only have a small sample size, about 40 subjects in both data sets, so a simple classifier performs better than a complex one.

For AD and MCI data sets, the Gaussian kernel performs better than linear kernel. The best error rate of AD for Gaussian kernel is 0.28 for $M = 10$ and $K = 10$ and the linear kernel gets 0.315 for the same M and K. The best error rate of MCI is 0.307 and the best error rate for linear kernel is 0.37. Because for AD and MCI data sets, we have much more subjects than in SLE and CPP studies, so the data can support a more complex classifier than the linear classifier.

In the linear kernel case, if the $M = 5$, then including more cubes will not enhance the performance of the classifier. For $M = 10$, including 20 or 50 cubes give similar classification error rate. This is not the case in the linear kernel. For Gaussian kernel, including more cubes will not increase the error rates. So Gaussian kernel can combine more kernels together, supporting a more complicated classifier.

For the individual region error map, $C_{gauss} = 1$ produces a classifier which overfits the data. But in the MKL case, $C_{gauss} = 1$ generally gets better results than $C_{gauss} =$ 10 and 50. In the MKL cases, we have more information of the class which can be used to fit a more complicated classifier.

3.6 Discussion

This chapter covers the analysis of structural MRI. VBM and SVM are the most popular methods in the structural MRI analysis. VBM is a voxel-based method which takes one voxel at a time and ignores the interaction between the voxels. SVM is a multivariate method which aligns all the voxels as a long vector and ignores the physical location of each voxel.

We propose a two-step procedure, taking both the interaction and the location information into consideration. The new method uses the multiple kernel learning technique which builds sub-kernels on different variables and then combines them in the second step. In our two-step procedure, we first select the informative regions based on the single kernel SVM analysis and then train a multiple kernel SVM to

combine the informative regions.

We test the performance of the two-step procedure in a simulation study. The study shows that the performance of the multiple kernel classifier is related to several factors, such as the strength of the signal, the level of the spatial correlation in the data and the number of kernels taken by the multiple kernel classifier.

In the real data analysis, we compare the two-step procedure to the traditional single kernel SVM in four data sets. The two-step procedure can achieve a lower classification error rate in all four cases. This suggests that using the multiple kernel SVM to allow a more flexible classifier can improve the performance.

We also test our method on few informative regions that are selected according to extra knowledge about the disease. Starting from few informative regions can reduce the misclassification error. Moreover we compare the results of the Gaussian kernel and the linear kernel. The linear kernel performs better in the cases with small sample size. As the sample size goes up, the data can support more complicated classifier.

CHAPTER IV

Conclusion and Future Work

This thesis discusses both the fMRI analysis and the structural MRI analysis. The former is a signal decomposition problem, decomposing the observed signals into different sources to identify the regions activated by the stimuli. The latter is a classification problem, classifying the subjects into the patients and the healthy controls to detect the regions affected by a disease. We propose new methods for both analyses and compare the results to the existing methods. We also discusses some possible directions for the future works.

4.1 fMRI Analysis

Chapter II covers the fMRI analysis. The fMRI signal is a series of brain scans recording the neural activities evoked by some stimuli. The purpose of the fMRI analysis is to identify the regions that respond to the stimuli. One challenge of the fMRI analysis is the high temporal and spatial correlation in the data. One popular method, GLM, uses an autoregressive model to model the temporal correlation. The autoregressive model assumes the same level of temporal correlation across the whole brain which is not a valid assumption. Instead of assuming a uniform correlation, we propose a Gaussian process model that allows different levels of correlation for different voxels. In the Gaussian process model, the observed signal is divided into three deterministic parts: a constant representing the scale of the time series, a linear function representing a linear trend usually observed in fMRI signals and a stimuli-related part measuring the strength of the neural activity and two random processes: a Gaussian process with Gaussian kernel measuring the temporal correlation of the signal and a white noise.

The estimation can either use a Bayesian approach, imposing a prior distribution on the parameters or use a frequentist approach through the EM algorithm. The simulation study shows that we can obtain reasonable estimates of the parameters using both methods. The real data analysis shows that different voxels in the brain

have different levels of temporal correlation. The correlation level is a smooth function across the whole brain. The activation areas identified by the new method is similar to the GLM method but with smoother boundaries.

Future Work

Gaussian process approach is a voxel-based method which means it analyzes one voxel at a time, ignoring the spatial correlation between the voxels. In future works, we can incorporate the spatial information into the covariance matrix of the Gaussian process. This means that the neighboring voxels are not modeled independently anymore. For the signal of voxel v_i at time t_s and the signal of voxel v_j at time t_k, the correlation is

$$\mathbf{Cor}(X_{\{v_i,t_s\}}, X_{\{v_j,t_k\}}) = f(v_i - v_j, t_s - t_k),$$

where $f(\tau, \nu)$ is a decreasing function in both τ and ν. In this case, we not only capture the temporal correlation but also model the spatial correlation of the fMRI signals.

GLM and ICA are two widely used methods for fMRI analysis. The popularity of GLM comes from its easy estimation and simple interpretation. The model specifies each component and estimates the parameters in explicit forms. However, GLM is a voxel-based method which does not take the spatial correlation into consideration. ICA is a multivariate methods which analyzes the whole brain together instead of one voxel at a time. However, ICA is an unsupervised method, with difficulties in the interpretation of those independent components.

We desire a multivariate supervised learning method that have the advantages from both GLM and ICA . Chapter II provides one possible solution, using a Gaussian process methods, dividing the signals into the deterministic experiment-related part

and the random Gaussian process part. The experiment-related part captures the signals of the neural activity and the Gaussian process models the temporal and spatial correlation in the fMRI scans. So designing a covariance matrix that can reflect both the temporal and spatial information is the key for the Gaussian process model.

4.2 Structural MRI Analysis

Chapter III covers the structural MRI analysis. The structural image is one scan of the whole brain with superb spatial resolution. It provides a detailed map of the brain, capturing the subtle changes brought by any effects such as diseases. The purpose of the structural MRI analysis is to classify the subjects into the patients and the healthy control group and to identify the regions that are affected by the diseases. One existing method, based on the machine learning technique SVM, treats all the voxels in the brain as a long vector. It ignores the local structures of the functional regions and the spatial correlation among the neighboring voxels. We propose a two-step procedure which takes the regional structure into consideration through the multiple kernel learning technique. The multiple kernel learning combines different kernels from different sources, allowing different variables contributing in different ways.

In the structural MRI analysis, we design the individual kernels on small regions in the brain. At the first step, the whole brain is divided into functional regions according to the AAL atlas of the human brain (*Fischl et al.*, 2002). A single kernel SVM classifier is trained on each region to measure the importance of each region. Then the top M regions with the lowest classification error rates are selected and further broken down into small $5 \times 5 \times 5$ cubes. A single kernel SVM is applied to the individual cubes to select the top K cubes with the lowest classification error rates. At the second step, the multiple kernel SVM builds a kernel on each cube first

and then combine them together. In this segmentation and combination steps, the multiple kernel learning achieves a more flexible classifier than the traditional SVM method.

Applying the two-step procedure on the four data sets shows that the new method can achieve a classifier better than the traditional SVM. For all the four data sets, the two-step procedure gets a lower error rate than using only the single kernel. Moreover, it can identify the disease-related regions that are confirmed by other works.

Future Work

For all the four data sets, both the traditional SVM and the multiple kernel SVM can not achieve a classification error rate good enough for the clinical diagnosis. One possible explanation lies in the preprocessing steps. The registration step maps all the brain scans into the same template, reducing the group differences between the patients and the healthy controls. In the future work, we can test the method on the unprocessed data.

In our real data analysis, we compare the performances between Gaussian kernel and the linear kernel. The results show that for small data set, about 40 subjects, the linear kernel can outperform Gaussian kernel. But for data set with 100 subjects, Gaussian kernel achieves a lower classification error rate than the linear kernel. The relation between the performance of different types of kernels and the sample size need further investigation.

In our method, we choose to select the informative region first and then select the informative cubes within the informative regions. We use the cubes of size $5 \times 5 \times 5$ as the smallest unit that our method can detect. There is a trade off between the size of the unit and the stability of the method. For small unit, such as the individual voxel, the method can identify very small area but with a lot of false positive voxels. For large unit, such as using the AAL regions to train the SVM, the method can

get stable results but very coarse resolution. The size of the unit for multiple kernel method also is a direction of further research.

BIBLIOGRAPHY

BIBLIOGRAPHY

ACOG Committee on Practice Bulletins–Gynecology. (2004), Acog practice bulletin no. 51. chronic pelvic pain., *Obstet Gynecol.*, *103*(3), 589–605.

Aguirre, G. K., E. Zarahn, and M. D'Esposito (1998), The variability of human, bold hemodynamic responses, *NeuroImage*, *8*(4), 360–369.

Appenzeller, S., B. Pike, L. Rittner, G. Leonard, M. Veilleux, and A. E. Clarke (2009), Thalamic volumes predict cognitive impairment evaluated by speed processing tasks in systemic lupus erythematosus [abstract], in *Arthritis Rheum 2009*, vol. 60 Suppl.

As-Sanie, S., R. Harris, V. Napadow, J. Kim, G. Neshewat, A. Kairys, D. Williams, D. Clauw, and T. Schmidt-Wilcke (2012), Changes in regional gray matter volume in women with chronic pelvic pain: a voxel-based morphometry study., *Pain.*, *153*(5), 1006–14.

Ashburner, J., and K. Friston (1999), Nonlinear spatial normalization using basis functions, *Human Brain Mapping*, *7*(4), 254–266.

Ashburner, J., and K. Friston (2005), Unified segmentation, *NeuroImage*, *26*, 839–851.

Ashburner, J., and K. J. Friston (2000), Voxel-based morphometrythe methods, *Neuroimage*, *11*, 805–821.

Ashburner, J., P. Neelin, D. L. Collins, A. C. Evans, and K. Friston (1997), Incorporating prior knowledge into image registration, *NeuroImage*, *6*, 344–352.

Asllani, I., C. Habeck, N. Scarmeas, A. Borogovac, T. R. Brown, and Y. Stern (2007), Multivariate and univariate analysis of continuous arterial spin labeling perfusion MRI in Alzheimer's disease, *J Cereb Blood Flow Metab*, *28*(4), 725–736.

Bell, A. J., and T. J. Sejnowski (1995), An information-maximisation approach to blind separation and blind deconvolution, *Neural Computation*, *7*, 1129–1159.

Benjamini, Y., and Y. Hochberg (1995), Controlling the false discovery rate: A practical and powerful approach to multiple testing, *Journal of the Royal Statistical Society. Series B (Methodological)*, *57*(1), 289–300.

Biswal, B. B., and J. L. Ulmer (1999), Blind source separation of multiple signal sources of fmri data sets using independent component analysis., *Journal of computer assisted tomography*, *23*(2), 265–271.

Bottino, C. M., C. C. Castro, R. Gomes, C. Buchpiguel, R. Marchetti, and M. Neto (2002), Volumetric mri measurements can differentiate alzheimer's disease, mild cognitive impairment, and normal aging., *Int Psychogeriatr.*, *14*(1), 59–72.

Boynton, G. M., S. A. Engel, G. H. Glover, and D. J. Heeger (1996), Linear systems analysis of functional magnetic resonance imaging in human v1, *The Journal of Neuroscience*, *16*, 4207–4221.

Cagnoli, P., P. Sundgren, A. Kairys, C. Graft, D. Clauw, S. Gebarski, W. McCune, and T. Schmidt-Wilcke (2012), Changes in regional brain morphology in neuropsychiatric systemic lupus erythematosus., *J Rheumatol.*, *39*(5), 959–67.

Calhoun, V., T. Adali, G. Pearlson, and J. Pekar (2001), Spatial and temporal independent component analysis of functional mri data containing a pair of task-related waveforms, *Human Brain Mapping*, *13*(1), 43–53.

Calhoun, V. D., T. Adali, L. K. Hansen, J. Larsen, and J. J. Pekar (2003), Ica of functional mri data: An overview, in *in Proceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation*, pp. 281–288.

Cardoso, J.-F. (1997), Infomax and maximum likelihood for blind source separation, *Signal Processing Letters, IEEE*, *4*(4), 112–114.

Cardoso, J.-F., and A. Souloumiac (1993), Blind beamforming for non gaussian signals, *IEE Proceedings-F*, *140*, 362–370.

Chan, D., et al. (2001), Patterns of temporal lobe atrophy in semantic dementia and Alzheimer's disease., *Annals of neurology*, *49*(4), 433–442.

Comon, P. (1994), Independent component analysis, a new concept?, *Signal Processing*, *36*(3), 287–314.

Cox, D. D., and R. Savoy (2003), fMRI 'Brain Reading': detecting and classifying distributed patterns of fMRI activity in human visual cortex, *NeuroImage*, *19*(2), 261–270.

Davatzikos, C. (2004), Why voxel-based morphometric analysis should be used with great caution when characterizing group differences, *Neuroimage*, *23*(1), 17–20.

Davis, R. (2001), *Encyclopedia of Environmetrics*, chap. Section on Stochastic Modeling and Environmental Change, Wiley.

Delfosse, N., and P. Loubaton (1995), Adaptive blind separation of independent sources: A deflation approach, *Signal Processing*, *45*(1), 59 – 83.

Draganski, B., C. Gaser, V. Busch, G. Schuierer, U. Bogdahn, and A. May (2004), Neuroplasticity: changes in grey matter induced by training., *Nature*, *427*(6972), 311–312.

Fan, Y., D. Shen, and C. Davatzikos (2005), Classification of structural images via high-dimensional image warping, robust feature extraction, and svm, in *Proceedings of the 8th international conference on Medical Image Computing and Computer-Assisted Intervention - Volume Part I*, MICCAI'05, pp. 1–8, Springer-Verlag.

Fan, Y., et al. (2007), Multivariate examination of brain abnormality using both structural and functional mri., *NeuroImage*, *36*(4), 1189–99.

Fischl, B., et al. (2002), *Neuron*, *33*(3), 341–355.

Friston, K., J. Ashburner, C. Frith, J. Poline, J. D. Heather, and R. Frackowiak (1995a), Spatial registration and normalization of images, *Human Brain Mapping*, *2*, 165–189.

Friston, K., A. Holmes, K. Worsley, J. Poline, C. Frith, and R. Frackowiak (1995b), Statistical parametric maps in functional imaging: A general linear approach, *Human Brain Mapping*, *2*, 189–210.

Friston, K., S. Williams, R. Howard, R. Frackowiak, and R. Turner (1996a), Movement-related effects in fMRI time-series, *Magnetic Resonance in Medicine*, *35*, 346–355.

Friston, K., J. Ashburner, S. Kiebel, T. Nichols, and W. Penny (Eds.) (2007), *Statistical Parametric Mapping: The Analysis of Functional Brain Images*, Academic Press.

Friston, K. J., S. Williams, R. Howard, R. S. Frackowiak, and R. Turner (1996b), Movement-related effects in fMRI time-series., *Magnetic resonance in medicine*, *35*(3), 346–355.

Genovese, C. R., N. A. Lazar, and T. Nichols (2002), Thresholding of statistical maps in functional neuroimaging using the false discovery rate, *NeuroImage*, *15*(4), 870 – 878.

Gitelman, D. R., J. Ashburner, K. J. Friston, L. K. Tyler, and C. J. Price (2001), Voxel-based morphometry of herpes simplex encephalitis., *NeuroImage*, *13*, 623–631.

Gönen, M., and E. Alpaydin (2011), Multiple kernel learning algorithms, *J. Mach. Learn. Res.*, *12*, 2211–2268.

Gosche, K. M., J. A. Mortimer, C. D. Smith, W. R. Markesbery, and D. A. Snowdon (2002), Hippocampal volume as an index of Alzheimer neuropathology, *Neurology*, *58*(10), 1476–1482.

Greenstein, D., B. Weisinger, J. D. Malley, L. Clasen, and N. Gogtay (2012), Using multivariate machine learning methods and structural mri to classify childhood onset schizophrenia and healthy controls, *Frontiers in Psychiatry*, *3*.

Huettel, S. A., A. W. Song, and G. McCarthy (2009), *Functional Magnetic Resonance Imaging, Second Edition*, Massachusetts: Sinauer.

Hyvärinen, A. (1998), New approximations of differential entropy for independent component analysis and projection pursuit, in *NIPS '97: Proceedings of the 1997 conference on Advances in neural information processing systems 10*, pp. 273–279, MIT Press.

Hyvärinen, A. (1999), Survey on independent component analysis, *Neural Computing Surveys*, *2*, 94–128.

Hyvärinen, A., and E. Oja (2000), Independent component analysis: algorithms and applications, *Neural Networks*, *13*(4-5), 411–430.

Jones, M. C., and R. Sibson (1987), What is projection pursuit?, *Journal of the Royal Statistical Society. Series A (General)*, *150*(1), 1–37.

Josephs, O., and R. N. A. Henson (1999), Event-related functional magnetic resonance: modelling, inference and optimization, *Phil. Trans. R. Soc. Lond. B*, *354*, 1215–1228.

Josephs, O., R. Turner, and K. Friston (1997), Event-related fmri, *Human Brain Mapping*, *5*(4), 243–248.

Karas, G., et al. (2003), A comprehensive study of gray matter loss in patients with alzheimer's disease using optimized voxel-based morphometry., *NeuroImage*, *18*(4), 896–907.

Kawasaki, Y., et al. (2007), Multivariate voxel-based morphometry successfully differentiates schizophrenia patients from healthy controls, *NeuroImage*, *34*(1), 235–242.

Keller, S. S., and N. Roberts (2009), Measurement of brain volume using mri: software, techniques, choices and prerequisite, *Journal of Anthropological Science*, *97*, 127–151.

Kinser, P. A., and P. Grobstein (2000), Brain structures and their funcitions.

Klöppel, S., et al. (2008), Automatic classification of MR scans in Alzheimer's disease, *Brain*, *131*(3), 681–689.

Kopelman, M., et al. (2001), Structural mri volumetric analysis in patients with organic amnesia, 2: correlations with anterograde memory and executive tests in 40 patients, *J Neurol Neurosurg Psychiatry.*, *71*(1), 23–28.

Kubicki, M., M. Shenton, D. Salisbury, Y. Hirayasu, K. Kasai, R. Kikinis, F. Jolesz, and R. McCarley (2002), Voxel-based morphometric analysis of gray matter in first episode schizophrenia., *NeuroImage*, *17*(4), 1711–9.

LaConte, S., S. Strother, V. Cherkassky, J. Anderson, and X. Hu (2005), Support vector machines for temporal classification of block design fMRI data, *Neuroimage*, *26*, 317–329.

Lanckriet, G. R. G., N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan (2004), Learning the kernel matrix with semidefinite programming, *J. Mach. Learn. Res.*, *5*, 27–72.

Lao, Z., D. Shen, Z. Xue, B. Karacali, S. M. Resnick, and C. Davatzikos (2004), Morphological classification of brains via high-dimensional shape transformations and machine learning methods., *Neuroimage*, *21*(1), 46–57.

Lyoo, I., M. Kim, A. Stoll, C. Demopulos, A. Parow, S. Dager, S. Friedman, D. Dunner, and P. Renshaw (2004), Frontal lobe gray matter density decreases in bipolar i disorder, *Biol Psychiatry*, *15*(55), 648–51.

Magnin, B., L. Mesrob, S. Kinkingnhun, M. Plgrini-Issac, O. Colliot, M. Sarazin, B. Dubois, S. Lehricy, and H. Benali (2009), Support vector machine-based classification of alzheimer's disease from whole-brain anatomical mri., *Neuroradiology*, *51*(2), 273–283.

Maguire, E. A., D. G. Gadian, I. S. Johnsrude, C. D. Good, J. Ashburner, R. S. J. Frackowiak, and C. D. Frith (2000), Navigation-related structural change in the hippocampi of taxi drivers, *Proceedings of the National Academy of Sciences*, *97*(8), 4398–4403.

McKeown, M., L. K. Hansen, and T. J. Sejnowski (2003), Independent component analysis for fmri: What is signal and what is noise?, *Current Opinion in Neurobiology*, *13*(5), 620–629.

McKeown, M. J., S. Makeig, G. G. Brown, T. P. Jung, S. S. Kindermann, A. J. Bell, and T. J. Sejnowski (1998), Analysis of fmri data by blind separation into independent spatial components., *Hum Brain Mapp*, *6*(3), 160–188.

Mechelli, A., C. Price, K. Friston, and J. Ashburner (2005), Voxel-based morphometry of the human brain: Methods and applications, *Current Medical Imaging Reviews*, pp. 105–113.

Moulton, E. A., J. D. Schmahmann, L. Becerra, and D. Borsook (2010), The cerebellum and pain: Passive integrator or active participator ?, *Brain Research Review*, *65*(1), 14–27.

Mueller, S. G., M. W. Weiner, L. J. Thal, R. C. Petersen, C. Jack, W. Jagust, J. Q. Trojanowski, A. W. Toga, and L. Beckett (2005), The alzheimer's disease neuroimaging initiative., *Neuroimaging clinics of North America*, *15*(4), 867–77.

Neal, R. M. (1998), Regression and classification using Gaussian process priors (with discussion), *Bayesian Statistics*, *6*, 475–501.

Noble, W. S. (2004), *Support vector machine applications in computational biology*, chap. 3, Computational molecular biology, MIT Press.

Ogawa, S., T.-M. Lee, A. S. Nayak, and P. Glynn (1990), Oxygenation-sensitive contrast in magnetic resonance image of rodent brain at high magnetic fields, *Magnetic Resonance in Medicine*, *14*(1), 68–78.

Oh, D., S. Kim, S. Jung, Y. Sung, S. Bang, S. Bae, and Y. Choi (2011), Precuneus hypoperfusion plays an important role in memory impairment of patients with systemic lupus erythematosus., *J Rheumatol.*, *20*(8), 855–60.

Pennanen, C., et al. (2005), A voxel based morphometry study on mild cognitive impairment, *Journal of Neurology, Neurosurgery & Psychiatry*, *76*(1), 11–14.

Petersen, K., L. Hansen, T. Kolenda, E. Rostrup, and S. Strother (2000), On the independent components in functional neuroimages, in *In Second International Symposium on Independent Component Analysis and Blind Signal Separation*, pp. 615–620.

Petersen, R. C., G. E. Smith, S. C. Waring, R. J. Ivnik, E. Kokmen, and E. G. Tangelos (1997), Aging, memory, and mild cognitive impairment, *International Psychogeriatrics*, *9*(Supplement S1), 65–69.

Pham, D. T., and P. Garat (1997), Blind separation of mixture of independent sources through a maximum likelihood approach, in *In Proc. EUSIPCO*, pp. 771–774.

Phillips, C. L., et al. (2011), Relevance vector machine consciousness classifier applied to cerebral metabolism of vegetative and locked-in patients, *NeuroImage*, *56*, 797–808.

Price, S., D. Paviour, R. Scahill, J. Stevens, M. Rossor, A. Lees, and N. Fox (2004), Voxel-based morphometry detects patterns of atrophy that help differentiate progressive supranuclear palsy and parkinson's disease., *NeuroImage*, *23*(2), 663–9.

Purdon, P. L., and R. M. Weisskoff (1998), Effect of temporal autocorrelation due to physiological noise and stimulus paradigm on voxel-level false-positive rates in fmri, *Human Brain Mapping*, *6*(4), 239–249.

Rajapakse, J. C., F. Kruggel, J. M. Maisog, and D. Y. von Cramon (1998), Modeling hemodynamic response for analysis of functional mri time-series, *Human Brain Mapping*, *6*(4), 283–300.

Rasmussen, C. E., and C. Williams (2006), *Gaussian Processes for Machine Learning*, MIT Press.

Roy, C., and C. Sherrington (1890), On the regulation of the blood-supply of the brain, *Journal of Physiology*, *11 (1-2)*, 85–158.17.

Salmond, C., J. Ashburner, F. Vargha-Khadem, A. Connelly, D. G. Gadian, and K. Friston (2002), The precision of anatomical normalisation in the medial temporal lobe using spatial basis functions, *NeuroImage*, *17*(1), 507–512.

Schroeter, M. L., T. Stein, N. Maslowski, and J. Neumann (2009), Neural correlates of alzheimer's disease and mild cognitive impairment: A systematic and quantitative meta-analysis involving 1351 patients, *NeuroImage*, *47*(4), 1196–1206.

Shenton, M. E., R. Kikinis, R. W. McCarley, D. Metcalf, J. Tieman, and F. A. Jolesz (1991), Application of automated mri volumetric measurement techniques to the ventricular system in schizophrenics and normal controls, *Schizophrenia Research*, *5*(2), 103 – 113.

Sonnenburg, S., G. Rätsch, C. Schäfer, and B. Schölkopf (2006), Large scale multiple kernel learning, *J. Mach. Learn. Res.*, *7*, 1531–1565.

Talairach, J., and P. Tournoux (1988), *Co-Planar Stereotaxic Atlas of the Human Brain: 3-D Proportional System: An Approach to Cerebral Imaging (Thieme Classics)*, Thieme.

Thieben, M., A. Duggins, C. Good, L. Gomes, N. Mahant, F. Richards, E. McCusker, and R. Frackowiak (2002), The distribution of structural neuropathology in preclinical huntington's disease., *Brain*, *125*(Pt8), 1815–28.

Tzourio-Mazoyer, N., B. Landeau, D. Papathanassiou, F. Crivello, O. Etard, N. Delcroix, B. Mazoyer, and M. Joliot (2002), Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain, *NeuroImage*, *15*(1), 273–289.

Van Hoesen, G., J. Augustinack, J. Dierking, S. Redman, and R. Thangavel (2000), The parahippocampal gyrus in alzheimer's disease. clinical and preclinical neuroanatomical correlates., *Ann N Y Acad Sci.*, *911*, 254–74.

Vapnik, V. N. (1995), *The nature of statistical learning theory*, Springer-Verlag New York, Inc., New York, NY, USA.

Whitwell, J., and C. J. Jack (2005), Comparisons between alzheimer disease, frontotemporal lobar degeneration, and normal aging with brain mapping, *Top Magn Reson Imaging*, *Dec 16(6)*, 409–25.

Wolz, R., V. Julkunen, J. Koikkalainen, E. Niskanen, D. P. Zhang, D. Rueckert, H. Soininen, J. Ltjnen, and the Alzheimer's Disease Neuroimaging Initiative (2011), Multi-method analysis of mri images in early diagnostics of alzheimer's disease, *PLoS ONE*, *6*(10), e25,446.

Woolrich, M. W., B. D. Ripley, M. Brady, and S. M. Smith (2001), Temporal autocorrelation in univariate linear modeling of FMRI data., *NeuroImage*, *14*(6), 1370–1386.

Worsley, K., S. Marrett, P. Neelin, A. C. Vandal, K. Friston, and A. C. Evans (1996), A unified statistical approach for determining significant voxels in images of cerebral activation, *Human Brain Mapping*, *4*, 58–73.

Wright, I., P. McGuire, J. Poline, J. Travere, R. Murray, C. Frith, R. Frackowiak, and K. Friston (1995), A voxel-based method for the statistical analysis of gray and white matter density applied to schizophrenia, *NeuroImage*, *2*, 244–252.

Zarahn, E., G. Aguirre, and M. D'Esposito (1997), A trial-based experimental design for fmri, *NeuroImage*, *6*(2), 122 – 138.