# ADVANCED STATISTICAL LEARNING TECHNIQUES FOR HIGH-DIMENSIONAL IMAGING DATA

Leo Yu-Feng Liu

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Statistics and Operations Research.

Chapel Hill
2018

Approved by:

Yufeng Liu

Hongtu Zhu

Steve Marron

Martin Styner

Kai Zhang

# ABSTRACT

LEO YU-FENG LIU: Advanced Statistical Learning Techniques for High-Dimensional Imaging Data
(Under the direction of Yufeng Liu and Hongtu Zhu)

With the rapid development of neuroimaging techniques, scientists are interested in identifying imaging biomarkers that are related to different subtypes or transitional stages of various cancers, neuropsychiatric diseases, and neurodegenerative diseases. Scalar-on-image models have been proven to demonstrate good performance in such tasks. However, due to their high dimensionality, traditional methods may not work well in the estimation of such models. Some existing penalization methods may improve the performance but fail to take the complex spatial structure of the neuroimaging data into account. In the past decade, the spatially regularized methods have been popular due to their good performance in terms of both estimation and prediction. Despite the progress, many challenges still remain. In particular, most existing image classification methods focus on binary classification and consequently may underperform for the tasks of classifying diseases with multiple subtypes or transitional stages. Moreover, neuroimaging data usually present significant heterogeneity across subjects. As a result, existing methods for homogeneous data may fail. In this dissertation, we investigate several new statistical learning techniques and propose a Spatial Multi-category Angle based Classifier (SMAC), a Subject Variant Scalar-on-Image Regression (SVSIR) model and a Masking Convolutional Neural Network (MCNN) model to address the above issues. Extensive simulation studies and practical applications in neuroscience are presented to demonstrate the effectiveness of our proposed methods.

# ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to the people who stood by me during my years at UNC. Without their support, this dissertation would not have been completed.

Firstly, I would like to thank my Ph.D. advisors, Professor Yufeng Liu and Professor Hongtu Zhu, for their patient guidance, continued encouragement and constructive advice throughout my time as their student. I have learned many good things from them both in academic work and personality. It is my greatest pleasure and honor to be their student. I must convey my sincere thanks to my dissertation committee members: Professor Steve Marron, Professor Martin Styner, and Professor Kai Zhang, for their time, support, and extremely valuable suggestions on my dissertation. I am also very grateful to the Department of Statistics and Operations Research for offering me the opportunity to study here five years ago, and providing great resources for my research. Last but not least, I am very thankful to my classmates, friends and my family for their generous support, great friendship and unconditional love.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# CHAPTER 1
# Introduction

## 1.1 Background

Neuroimaging technology has been rapidly developed in the past decades. Many imaging techniques are widely used to unravel the mystery about the structure and functionality of our neural system, and provide valuable information for the diagnosis and treatment of certain diseases (Giedd et al., 1999; Ogawa et al., 1990; Khoo et al., 1997). For example, advanced imaging techniques, including Computed Tomography (CT), Magnetic Resonance Imaging (MRI) and Positron Emission Tomography (PET) are commonly used both in clinical applications and scientific research to study the functionality of human brains. Figure 1.1 displays some typical brain images, obtained using such techniques. They create images in different ways and measure different aspects of the brain structure and activities. A CT scan uses X-rays taken from different angles to produce cross-sectional images that measure different levels of tissue density inside the brain. MRI is based on the science of nuclear magnetic resonance and uses the gradient field of the radio-frequency signal of hydrogen atoms nuclei to generate brain images. PET images measure the metabolic processes, such as flows of blood to different parts of the brain, via detecting the radioactivity of the injected tracer.

Neuroimaging techniques are widely used for imaging-guided diagnosis procedures. In general, such procedures include three steps: image acquisition, image processing, and diagnosis. An example of MRI-aided diagnosis is illustrated in Figure 1.2. The imaging-guided diagnosis can significantly improve the accuracy of diagnosed results but requires the expertise of well-trained radiologists, which can be expensive to obtain in practice.

The computer-aided diagnosis (CAD) is the system that assists radiologists in the interpretation of medical images. This can potentially expedite the procedure of imaging-guided diagnosis. Currently, CAD techniques have been widely used in screening breast cancer in the preventive

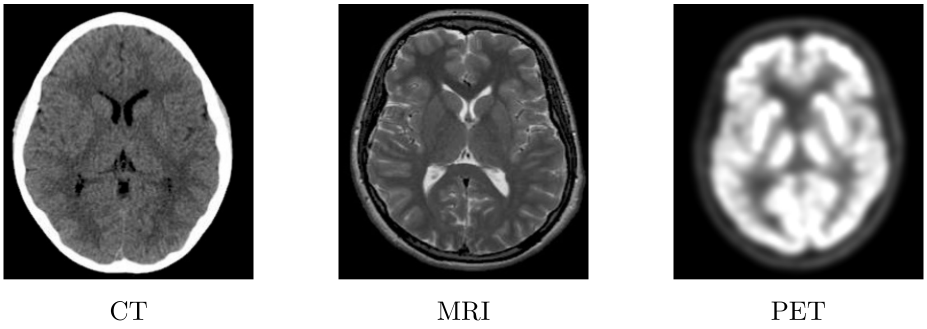


**Figure 1.1:** Plots of three popular modalities of human brain images: Computed Tomography (CT) in the left panel, Magnetic Resonance Imaging (MRI) in the middle panel and Positron Emission Tomography (PET) in the right panel. All images are displayed in the transverse direction.

medical check-ups in mammography, and in the detection of tumors in the CT scans of lung cancer patients.

The key concept of CAD is to build efficient statistical models, that use medical imaging data to predict important clinical information, and eventually assist radiologists to make diagnosis decisions. There are two types of supervised learning problems in the field of CAD, classification and regression. These two problems can be solved by scalar-on-image models, whose response represents scalar variables and covariate corresponds to imaging data. For classification problems, the scalar responses represent class labels that indicate different stages of disease development, or different clinically meaningful subgroups of patients. For regression, the models are used to predict certain clinical scores as continuous variables, based on image covariates. These clinical scores are usually highly related to the pathology of diseases and can be used as an important guideline to evaluate the effectiveness of treatment plans. For instance, the MiniMental State Examination

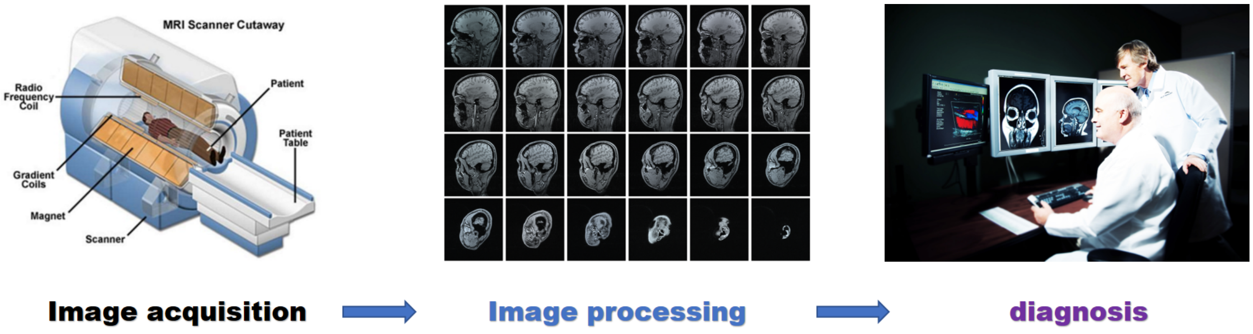


**Figure 1.2:** A typical imaging assisted diagnosis procedure.

(MMSE) score is commonly considered as a benchmark in clinical and research settings to evaluate cognitive impairment and to screen for dementia (Pangman et al., 2000).

For the rest of this chapter, we briefly introduce the framework of scalar-on-image models and review some related works in the literature.

### 1.2 Data structure and notation

We use bold symbols to represent image variables, such as $\boldsymbol{X}$ and use regular symbols for scalar and other non-image variables, such as $y$. The notation $[n]$ represents the set $\{1, 2, \ldots, n\}$. The data in the scalar-on-image model are given in pairs of $(\boldsymbol{X}_i, y_i)$'s, where $y_i$ denotes the scalar response and $\boldsymbol{X}_i$ represents the corresponding covariate image for the $i$-th subject.

The image variable $\boldsymbol{X}$ can be viewed as a real-valued function over a bounded image domain, i.e., $\boldsymbol{X} = \{\boldsymbol{X}(t) \in \mathbb{R};\ \forall t \in \mathcal{D}\}$. Here the image domain $\mathcal{D}$ denotes a bounded 2-D surface or 3-D volume, and $t$ is the corresponding location index, which can be a vector of length 2 or 3 according to the dimension of $\mathcal{D}$. The function value $\boldsymbol{X}(t)$ can be the raw image intensity or some other measurement at location $t$.

In practice, digital images are often collected with finite resolution, and in this case, the image functions are only evaluated at certain grid points of the whole image domain. Those grid points are usually referred as pixels/voxels. The associated imaging data $\boldsymbol{X}$ are then presented as 2-D matrices or 3-D tensors, with each entry as the pixel/voxel value.

In general, a scalar-on-image model is defined as follows,

$$y_i = f(\langle \boldsymbol{X}_i, \boldsymbol{\beta} \rangle); \qquad \text{for } i = 1, \ldots, n, \tag{1.1}$$

where $\boldsymbol{\beta}$ denotes the coefficient image corresponding to $\boldsymbol{X}_i$'s, which can also be treated as a function over the same image domain $\mathcal{D}$. The operation $\langle \cdot, \cdot \rangle$ denotes the inner product of the two images, which is given by

$$\langle \boldsymbol{X}_i, \boldsymbol{\beta} \rangle = \int_{t \in \mathcal{D}} \boldsymbol{X}_i(t) \boldsymbol{\beta}(t) dt. \tag{1.2}$$

Note that, in the discrete setting, the inner product in (1.2) is equivalent to the inner product of two matrices or tensors for 2-D or 3-D images respectively.

According to the definition in Equation (1.1), the scalar-on-image model basically assumes that the $y_i$'s respond to the change of covariate images $\boldsymbol{X}_i$' through the inner product of $\boldsymbol{X}_i$ and $\boldsymbol{\beta}$, defined in Equation (1.2). This assumption guarantees the coefficient image $\boldsymbol{\beta}$ lies on the same space as the covariate images, and thus can incorporate the spatial information of the image domain. We will discuss this property in detail in Section 2.2.3.

In the machine learning literature, the estimation of scalar-on-image models can be summarized in the *loss + penalty* framework (Hastie et al., 2005). The loss function is used to ensure the goodness-of-fit of the model on the training data, and the penalty term is introduced to avoid overfitting and encourage some desired structure in the estimated coefficients, such as sparsity. Under this framework, the challenges in estimating the scalar-on-image models mainly arise from the high dimensionality, complex spatial structure, and strong noise in imaging data. High dimensionality is a very common phenomenon in medical imaging data. For example, a typical MRI image of size $256 \times 256 \times 256$ corresponds to a variable in the $16,777,216$ dimensional space of the statistical model. Due to the high cost of image acquisition facilities, the sample size, i.e., the number of participants in neuroimaging studies is usually very small. This makes the problem fall into the high dimensional low sample size (HDLSS) realm, which was discussed by Hall et al. (2005) in detail. Furthermore, the inherent biological structure of the objects in medical images often present complex spatial correlation and smoothness. Without considering such structure, the models can be hard to interpret and underperformed in terms of prediction. Moreover, the noise of neuroimaging data can be generated in every step of the data acquisition. For example, head motions and machine vibrations in a MRI scan will blur the brain images, and the registration and alignment error in image processing will generate some systematic error of the spatial locations. These noises are spatially correlated and strongly impact the estimation and prediction accuracy, thus require special techniques to deal with.

In the literature, many sparse regularization techniques have been proposed to handle high-dimensional data, including imaging data as a special case. For instance, Tibshirani (1996) introduced the Lasso regularization by imposing an $L_1$ norm penalty to high dimensional least squares estimation. It can efficiently solve the dimensionality issue in most cases, but when dealing with groups of correlated predictive variables, this method tends to select only a few variables as representatives of the groups and ignores the rest. Zou and Hastie (2005) proposed the "Elastic-Net" penalty which regularizes both $L_1$ and $L_2$ norms of the coefficients. It can avoid the selection issue with correlated variables by encouraging a grouping effect.

In general, the sparse regularization methods perform the variable estimation and selection simultaneously. They can improve both estimation and prediction performances in a general high dimensional setting. However, for imaging data, the predictive variables are not only sparse, but also spatially clustered in the image domain. Without considering such spatial structures, these regularization methods may underperform when applied in scalar-on-image models. In particular, they tend to delivery coefficient images containing only isolated voxels, which are less clinically meaningful.

To effectively handle imaging data, it is critically important to incorporate its spatial smoothness and correlation structure. One efficient approach is to impose the spatial regularization penalty. It has been proven that this approach helps to deliver interpretable coefficient images and improve prediction performance. For instance, Rudin et al. (1992) introduced the total variation penalty that controls the differences between intensities of the adjacent pixels/voxels in the coefficient image. Recently, Grosenick et al. (2013) proposed a spatial smoothing penalty named GraphNet by incorporating local graph structure into the Elastic-Net (Zou and Hastie, 2005) regularization. Both methods yield spatially clustered signals in the coefficient images. The total variation penalty yields more clear boundaries between zero and non-zero regions, while GraphNet achieves more spatial smoothness. Despite of progress in these methods, there are still many unsolved problems in the scalar-on-image models. We will discuss some of the unique challenges for image based classifications and regressions separately in the following sections.

### 1.3 Image based classification

Image based classification is of great interest in the field of neuroscience. It is expected that, for many diseases, such as Alzheimer's disease (AD) and breast cancer, medical images carry clinically relevant features associated with the pathophysiology of these diseases. Such features are usually referred as imaging biomarkers in neuroscience. An efficient image classification model should not only be able to classify patients into clinically meaningful subgroups, but also identify those relevant imaging biomarkers. This is critically important to improve the accuracy of the neuroimaging based computer-aided diagnosis (CAD) and possibly improve treatment plans at an early stage of the diseases.

The mathematical formulation of the image based classification problems can be expressed using the scalar-on-image model in Equation (1.1). The response $y_i$'s are defined as categorical variables, i.e. $y_i \in \{1, \ldots, K\}$, representing different groups of subjects. Here $K$ denotes the total number of classes, which is an integer much smaller than the sample size $n$. The classification model essentially assumes that the pairs of $(\boldsymbol{X}_i, y_i)$'s are drawn from an unknown distribution $\mathcal{P}(\boldsymbol{X}, y)$ defined over $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ denotes the space of all images on the spatial domain $\mathcal{D}$ and $\mathcal{Y} = \{1, \ldots, K\}$ defines the associated class label space. A classification rule $f : \mathcal{X} \to \mathcal{Y}$, is a function that maps covariate image $\boldsymbol{X}$ into the class label space $\{1, \ldots, K\}$. A natural criteria to evaluate a classification rule is to use the corresponding classification error, i.e., $E_{\boldsymbol{X},y}\left[\mathbb{I}(f(\boldsymbol{X}) \neq y)\right]$, where $\mathbb{I}(\cdot)$ represents the indicator function. The optimal classification rule is denoted as the Bayes rule, which theoretically minimizes the classification error, i.e.,

$$\begin{aligned} f^*(X) &= \operatorname*{argmin}_f E_{\boldsymbol{X},y}\left[\mathbb{I}(f(\boldsymbol{X}) \neq y)\right] \\ &= \operatorname*{argmax}_y \mathbb{P}\left(y|\boldsymbol{X}\right). \end{aligned}$$

In practice, since the underlying distribution $\mathcal{P}(\boldsymbol{X}, y)$ is unknown, estimation of the classification rule is essentially finding the functions approximating the theoretical Bayes rule.

### 1.3.1 Binary classification

The simplest case of classification problems is binary classification, where the total number of classes is 2, i.e., $K = 2$. Many approaches have been proposed for this problem. Among various techniques, there are two important groups of classification methods: likelihood based and margin-based methods. The likelihood based methods try to solve the classification problem directly by modeling the distribution of the covariates and responses. Examples include but are not limited to Linear Discriminant Analysis (LDA) (Fisher, 1936), Quadratic Discriminant Analysis

(QDA) (Hastie et al., 2005) and logistic regression (Cox, 1958; Walker and Duncan, 1967; Hastie et al., 2005). In LDA, the underlying conditional distributions of the covariates, $P(\boldsymbol{X}|y=1)$ and $P(\boldsymbol{X}|y=2)$ are commonly assumed to be normally distributed with equal covariance matrices, while in QDA the two classes are allowed to have different covariance structure. Both methods use the maximum likelihood to estimate the conditional distributions, and apply the Bayes's theorem to predict the class labels. In logistic regression, the class labels are assumed to follow a Bernoulli distribution $Ber(p)$. The covariates determine the mean of the distribution through a link function $g(p) = \langle \boldsymbol{X}, \boldsymbol{\beta} \rangle$. The coefficient image $\boldsymbol{\beta}$ can be obtained using iterative reweighted least squares estimation. These methods work well for low dimensional data, but may underperform when dealing with imaging data, because the estimation procedure can be unstable in the high dimensional setting. Furthermore, the distribution assumptions may not hold for neuroimaging data.

In the past few decades, margin-based methods are getting more and more popular due to their flexibility and improved prediction performance. These methods provide a different view from the likelihood based approaches. Instead of imposing some distributional assumption, these methods directly estimate the classification boundary. In particular, the class label $y \in \{1, 2\}$ is coded as

$$W_y = \begin{cases} -1 & \text{if } y = 1 \\ +1 & \text{if } y = 2, \end{cases}$$

and a function $f(\cdot)$ is introduced, such that $sign(W_y f(\boldsymbol{X}))$ can be directly used as the classification rule. Among various margin-based classifiers, perhaps the most well known one is the Support Vector Machines (SVM) (Vapnik, 2013). It estimates the classification rule by solving the following optimization:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} l \left( W_{y_i} \langle \boldsymbol{X}_i \boldsymbol{\beta} \rangle \right) + \lambda \|\boldsymbol{\beta}\|^2,$$

where $l(u) = \max(0, 1-u)$ denotes the hinge loss, and $\lambda$ is a parameter controlling the tolerance level of misclassification. (Marron et al., 2007) pointed out that the SVM may suffer from the "data-piling" phenomena in the high-dimensional setting due to the non-differentiability of the hinge loss. They proposed a distance-weighted discrimination (DWD) method using a differentiable loss function to avoid this issue. Recently Liu et al. (2011) proposed a Large-margin Unified Machine

(LUM) which covers a range of the margin-based classifiers, including SVM and DWD as special cases. We will revisit LUM in Chapter 2 and discuss the details.

Binary classifiers have been widely used and well studied in the image based classification problems. We refer readers to Rathore et al. (2017) for a comprehensive review of the development on binary classifiers for imaging data in the past thirty years.

### 1.3.2 Multi-category classification

In contrast to the significant progress in binary image based classification, the developments in multi-category classification are quite limited in the neuroimaging literature. However, multi-category classification is of great importance and deserves more attention. In fact, many neurodegenerative diseases, such as Alzheimer's disease, often have multiple subtypes and transitional stages in their pathophysiological process. Only classifying patients into the disease and health control groups cannot provide sufficient information to characterize the pathophysiological progress. On the other hand, some diseases, such as breast cancer, may have multiple subtypes. Accurately identifying the subtypes of the disease can greatly improve the effect of personalized treatments.

There are different ways to extend binary classifiers for the multi-category classification. One approach is to conduct sequential binary classifications via the one-versus-one or one-versus-rest strategy. In this case, the multi-category problem is decomposed into $K(K-1)/2$ or $K$ binary classification problems respectively, and the prediction rule is determined by the majority vote. These methods have been proven to be suboptimal and yield ambiguous label assignments when there are no dominating classes (Liu and Yuan, 2011). Other classifiers solve the classification problem simultaneously by mapping the covariate to a vector with length equal to the total number of categories, i.e., $f(\boldsymbol{X}) \in \mathbb{R}^K$. The examples of such classifiers can be found in (Zhu and Hastie, 2005), (Zhu et al., 2009) and (Liu and Yuan, 2011). A sum-to-zero constraint on the predicted vector is usually applied to achieve desirable theoretical properties, but this may increase the complexity of the corresponding optimization. In Chapter 2, we propose a spatial angle-based classifier to efficiently solve these issues in multi-category neuroimaging classification.

### 1.4 Image based regression

Image based regression is another important application of the scalar-on-image models. The responses $y_i$'s in such models are continuous and may represent certain pathologically relevant clinical scores. For example, the MiniMental State Examination (MMSE) and Alzheimer's Disease Assessment Scale Cognitive (ADAS-Cog) scores are widely used to access the cognitive impairment of the patients with Alzheimer's disease. An effective imaging-based regression model should be able to accurately predict the clinical scores, and efficiently extract the informative imaging biomarkers from the data. This requires a one-to-one correspondence between the location indices in the coefficient images and the covariate images. Thus the linear models are commonly considered and formulated as follows,

$$y_i = \beta_0 + \langle \boldsymbol{X}_i, \boldsymbol{\beta} \rangle + \epsilon_i, \tag{1.3}$$

where $\epsilon_i$'s are the i.i.d. Gaussian noise with mean zeros and the finite variance $\sigma^2$.

Note that Model (1.3) can be regarded as a special case of functional linear regressions (FLR) if we treat the images as functions over the image domain. It can also be explained as an extension of the high-dimensional linear models (HDM) if the images are represented as the discrete pixel/voxel values. Both modeling frameworks are extensively studied in the literature among the past decade. We refer readers to the well-known monographs of Ramsay and Silverman (2005), Ferraty and Vieu (2006) and Bühlmann and Van De Geer (2011) for details. Despite of the flexibility of the FLR and HDM, the imaging-based regression models still have some unique challenges that cannot be solved by these two frameworks. For example, the HDM assumes the feature indices are interchangeable, but in imaging-based regression these indices are ordered according to the spatial location of the covariate images and thus not interchangeable.

Another major challenge for imaging-based regression models comes from strong heterogeneity among subjects. most existing methods do not work well to deal with such challenges. We will discuss the heterogeneity issue in detail in Chapter 3 and introduce some novel techniques to tackle the problem.

### 1.5 Deep convolutional neural network model

In the past few years, the deep convolutional neural network (CNN) models have raised huge attention by demonstrating very competitive performance in image related problems, including classification, detection and segmentation. For example, the CNN models have already beat humans in terms of classification accuracy in the MNIST digit recognition (LeCun et al., 1998; Wan et al., 2013) and ImageNet (Deng et al., 2009; He et al., 2016) classification problems. The application of deep CNN models in neuroimaging problems are also well studied in the recent literature, e.g., (Payan and Montana, 2015) used a 3D convolutional neural network with pre-trained sparse encoders to predict the Alzheimer's disease using the MRI images; (Wang et al., 2014) applied the extracted features from the CNN models to build a mitosis detector; (Moeskops et al., 2016) and (Zhang et al., 2015) proposed a method using deep CNN models to handle brain segmentation problems.

While the improvement of deep CNN models in image classification and segmentation problems is impressively significant, the model itself works as a "black box" in most cases. Many applications in neuroimaging are still based on existing networks. We will propose a novel convolutional neural network that can handle both the prediction and segmentation tasks simultaneously, and use the estimated segmentation results as a masking image that can indicate the regions in original images that are related to the prediction task.

### 1.6 New contributions and outline

In this dissertation, I focus on the predictive scalar-on-image models with application in neuroimaging studies. Both classification and regression problems are investigated. The major contributions include extending existing methods to the high dimensional neuroimaging setting and proposing new techniques that overcome some unique challenges in neuroimaging studies.

In Chapter 2, we propose a novel Spatial Multi-category Angle-based Classifier (SMAC) for neuroimaging data. The proposed method not only utilizes the spatial structure of high-dimensional imaging data but also handles both binary and multi-category classification problems. We also introduce an efficient and flexible algorithm based on an alternative direction method of multipliers (ADMM) algorithm to solve the large-scale optimization problem for SMAC and other similar regularized methods. Both our simulation and application in the Alzheimer's Disease Neuroimaging Initiative (ADNI) study demonstrate the usefulness of SMAC.

In Chapter 3, we investigate the scalar-on-image regression problems and propose a Subject Variant Scalar-on-Image Regression (SVSIR) model. The SVSIR can yield desired spatially smoothing and sparse coefficient images, and incorporate heterogeneity structure among the patients. Extensive numerical studies demonstrate the improvement in terms of both estimation and prediction performance. We also apply the proposed model in the ADNI study, to predict cognitive scores based on MRI data.

In Chapter 4, we propose a novel deep neural network model that can handle both segmentation and prediction simultaneously. More importantly, the segmentation module of the network can work as a regularization of the input image, and generate a mask that rules out the irrelevant regions of the input image and eventually improves the interpretability of the model and the prediction accuracy. We demonstrate the usefulness of the model using various datasets, including the MRI data from the ADNI study.

CHAPTER 2

# SMAC: Spatial Multi-category Angle based Classifier for High-dimensional Neuroimaging Data

## 2.1 Introduction

With advances in modern imaging technology, it is becoming increasingly prevalent to collect high-dimensional imaging data (e.g., magnetic resonance imaging [MRI]) in order to extract imaging biomarkers (or features) that are useful for various tasks, including disease detection, diagnosis, prognosis, and treatment, among many others (Chen et al., 1998; Lopez et al., 2009; Ramírez et al., 2009). For many diseases, such as Alzheimer's disease (AD) and breast cancer, it is expected that medical images contain clinically relevant information associated with their pathophysiology. A critical challenge is determining how to build a predictive model (or classifier) that can classify patients into clinically meaningful subgroups according to their imaging data. Such a model may improve the clinical care of these patients and possibly slow their disease progression.

In the current literature, there exist two groups of classification methods for imaging data, including feature-based analysis and image-based analysis. Feature-based analysis consists of (i) converting medical images into a set of features and (ii) building classifiers based on these extracted features. Standard feature extraction methods often extract some summary statistics (e.g., mean image intensity) in either segmented tumors or prefixed regions of interest (ROIs) in a template space. For example, Rusinek et al. (2004) used the partial volumes of the brain and cerebrospinal fluid (CSF) to classify AD versus normal control (NC), and Zhu et al. (2014) built a multi-category classifier using sparse linear discriminant analysis based on features extracted from 93 ROIs of both MRI and positron emission technology (PET) images. More examples of feature-based analysis can be found in Xu et al. (2000), Busatto et al. (2003), Colliot et al. (2008), and Yu et al. (2014). The major drawback of these feature-based methods is that they require knowledge of spatial segmentation to identify meaningful ROIs in order to extract informative, discriminating and independent features for the classification task.

Image-based analysis, however, uses raw imaging data across all grid points. Two key advantages of using raw imaging data include potential gain in classification accuracy and spatially interpretable coefficient maps of the classifiers in the original image space. The main challenges for image-based analysis include (i) high dimensionality, (ii) complex spatial information and (iii) noisy functional data. For example, a typical T1-weighted MR image of size $256 \times 256 \times 256$ will yield a $16,777,216$ dimensional space, and due to the inherent biological structure of the brain, these data also have complex spatial correlation and smoothness.

Many methods in the literature apply a pre-screening procedure to reduce the dimensionality of the imaging data, and build classifiers in the reduced image space. For example, Liu et al. (2012) applied the ensemble of multiple classifiers based on randomly selected patches of the MR images, and Hinrichs et al. (2011) built multiple kernel support vector machines based on 2,000 to 250,000 features selected by voxel-wise t-tests. The pre-screening procedure can significantly reduce the computational cost in estimating the classifiers, but potentially loses important predictive information. On the other hand, many regularization techniques have been proposed to directly handle high-dimensional data, including imaging data as a special case (Grosenick et al., 2008, 2009; Yamashita et al., 2008; Van Gerven and Heskes, 2012). For instance, Yamashita et al. (2008) proposed a method by imposing $L_2$ norm regularization to logistic regression for classification of functional MRI data in various tasks; whereas Casanova et al. (2011) applied elastic-net penalized regression to distinguish between patients with AD versus NCs based on both gray matter and white matter segmentation maps. These regularization methods perform simultaneous estimation of coefficients across all voxels and select the predictive voxels. Since most standard regularization methods do not account for the spatial structure of imaging data, their resulting classifiers usually contain only isolated voxels; thus, it can be difficult to interpret the results. Moreover, standard sparsity penalties, such as $L_1$, can be sub-optimal for the high-dimensional prediction problems considered here, since the effect of high-dimensional imaging data on certain categories is often spatially clustered and non-sparse.

To effectively handle imaging data, it is critically important to utilize the spatial smoothness and correlation of imaging data in the construction of classifiers. For instance, Grosenick et al. (2013)

proposed a spatial smoothing classifier based on the GraphNet penalty. Furthermore, Watanabe et al. (2014) developed a spatial support vector machine (SSVM) classifier based on the fused lasso (FL) and GraphNet penalties. These methods yield meaningful coefficient images and achieve good accuracy for binary neuroimaging classification, but are not directly applicable to multi-category classification problems.

The aim of this chapter is to develop a spatial multi-category angle-based classifier (SMAC) for high-dimensional imaging data. Compared with the existing methods in the literature, three major methodological contributions of this chapter are as follows:

- The proposed SMAC not only utilizes the spatial structure of images, but also extends the angle-based classification framework recently developed by Zhang and Liu (2014) to perform simultaneous multi-category classification of imaging data.

- We use a hybrid of a generalized total variation (TV) penalty (Tibshirani et al., 2005) and a sparse $L_1$ penalty, namely an FL penalty, to identify spatially aggregated clusters that are important for discriminating different classes. Our methods are able to deliver competitive classification accuracy and interpretable imaging biomarkers.

- We have developed the SMAC package by using both MATLAB and Python and will release it through the website "https://www.nitrc.org/". Our package includes a graphical user interface that is freely downloadable from the same website. Our SMAC package can handle 1-dimensional (1-D) curves, 2-dimensional (2-D) surfaces, and 3-dimensional (3-D) volumes.

The rest of the chapter is organized as follows. In Section 2, we introduce the SMAC framework and describe an optimization algorithm to efficiently estimate the model coefficients. We use two simulation experiments and the Alzheimer's Disease Neuroimaging Initiative (ADNI) data in Section 3 to examine the finite-sample performance of SMAC. In Section 4, we conclude with some discussion.

## 2.2 Methods and materials

### 2.2.1 Data Structure

One important classification problem in the neuroimaging literature is to predict the disease status of patients based on their neurological images. The class label is denoted by a categorical response variable $y$, usually taking values of $1, 2, \ldots, K$, indicating $K$ different classes of interest. The covariate $\boldsymbol{X} = \{x_d : d \in \mathcal{D}\} \in \mathbb{R}^p$ represents the observed imaging data, where $\mathcal{D}$ denotes the spatial space of the image, which can be a 1-D curve, 2-D surface or 3-D volume, and $d$ is a vector of length 1, 2 or 3, indicating the location of the corresponding voxel in the image. Without loss of generality, we focus on 3-D real valued images in this chapter, and use $p$ as the dimension of the imaging data, which equals the total number of voxels in the image.

### 2.2.2 Statistical Classification Framework

For a $K$-category classification problem, a statistical classifier builds a map from the covariate space $\mathbb{R}^p$ to the category space $\{1, \ldots, K\}$. Given a new observation $\boldsymbol{X}^*$, the classifier predicts the associated class label $y^*$ as $\hat{y}^*$. To build the classifier, many statistical procedures can be fitted into the regularization framework of *loss + penalty*. A loss function $l(\cdot)$ is introduced to ensure the goodness of fit of the resulting model to the training data. Two groups of loss functions that are commonly used in the literature include likelihood-based and margin-based loss functions. Likelihood-based methods usually impose some assumption of probability distributions on the data and then establish the classification rule by solving some parametric statistical models. Examples of these methods include Fishers linear discriminant analysis (LDA) (Fisher, 1936) and logistic regression (Hastie et al., 2005). In contrast, margin-based methods solve the classification problems without imposing a strong distributional assumption on the data. Specifically, a margin-based method uses a functional margin as the input of the loss function $l(\cdot)$. The values of the functional margins are directly associated with the accuracy of the class label assignment. For binary classification with the class label $W_y \in \{\pm 1\}$ for $y \in \{1, 2\}$, one can obtain a function $f(x)$ and use $\hat{W}_y = sign(f(\boldsymbol{X}))$ as a classification rule. In this case, the functional margin is defined as $W_y f(\boldsymbol{X})$, indicating the correctness of the classification. Our proposed classifier belongs to margin-based methods.

When dealing with high-dimensional data, a regularization term is usually added to the loss function to prevent the models from over-fitting the training data. The choice of the regularization term is based on prior knowledge of the data structure and the properties of the specific penalty. For instance, the $L_1$ norm penalty can be utilized to learn the sparse structure of data (Tibshirani, 1996), and the $L_2$ type of penalties encourage continuous shrinkage in the estimation (Zou and Hastie, 2005). To choose the penalty term for handling the neuroimaging data, it is necessary to account for its high dimensionality and complex image structure. A desired penalty should encourage sparsity, while incorporating the spatial structure of the imaging data.

#### 2.2.2.1 Binary Large-Margin Classifiers

Many "off the shelf" classifiers are potential candidates for neuroimaging classification. Examples range from the very classical LDA (Fisher, 1936) and logistic regression (Hastie et al., 2005) to the recent machine learning techniques, such as the support vector machine (Boser et al., 1992) and boosting (Friedman et al., 2000). The choice of the classifier depends on the data structure and the goal of classification. However, there is no clear guideline about which classifier to choose in each complicated case. (Liu et al., 2011) proposed a large-margin unified classifier (LUM), covering a rich family of classification methods, which allows us to tune our loss function within the rich LUM family to obtain a satisfactory solution. In this chapter, we choose a special LUM loss function which has the following form:

$$l(u) = \begin{cases} 1 - u, & \text{if } u < 0; \\ e^{-u}, & \text{if } u \geq 0. \end{cases} \tag{2.1}$$

This special loss can be viewed as a hybrid of the support vector machine and AdaBoost, which allows us to maximize the separation margin and dynamically assign weights in "weak" learners (Freund and Schapire, 1997). We refer readers to the original paper for further details of the LUM loss.

Despite the potential improvement in classification performance when using LUM, this classifier was originally proposed to solve binary classification problems. The extension to multi-category cases requires additional effort. We address this issue in the following section.

#### 2.2.2.2 Multi-category Large-margin Classifiers

To handle multi-category data, one simple approach is to conduct binary classification sequentially via the one-versus-one or one-versus-the-rest scheme in order to predict the class labels. These methods have been proven to be suboptimal when there is no dominating class (Liu and Yuan, 2011). Other classifiers solve the classification problem simultaneously by mapping covariates to a vector with the length equal to the total number of categories. Such classifiers can be found in (Zhu and Hastie, 2005), (Zhu et al., 2009) and (Liu and Yuan, 2011). A sum-to-zero constraint on the predicted vector is usually applied to achieve desirable theoretical properties, but may increase the complexity of the corresponding optimization. Without this constraint, (Zhang and Liu, 2014) proposed a multi-category angle-based classifier (MAC) that can achieve the Fisher consistency and some other desirable properties.

For a $K$-category classification problem ($K \geq 2$), MAC creates a map from the class labels $y \in [K]$ to the vertices of a regular simplex in the $(K-1)$-dimensional space, i.e.,

$$W_y = \begin{cases} (K-1)^{-1/2}\xi, & \text{if } y = 1; \\ -\frac{1+K^{1/2}}{(K-1)^{3/2}}\xi + \left(\frac{K}{K-1}\right)^{1/2} e_{y-1}, & \text{if } y \in [K]/1, \end{cases} \tag{2.2}$$

where $\xi \in \mathbb{R}^{K-1}$ is a vector with all elements being 1, and $e_y \in \mathbb{R}^{K-1}$ is a vector such that all elements are 0, except that the $y$-th component is 1. Note that for $K = 2$, it reduces to the traditional binary classification with labels $W_y \in \{\pm 1\}$. Due to the property of the regular simplexes, the angles between any two projected class labels are equal, i.e., $\angle(W_y, W_{y'}) = C_K$ for all $y \neq y'$.

Instead of directly using the original class label $y$, MAC uses the projected class label $W_y$ to solve the multi-category problem. In particular, we construct a function that maps the covariate $\boldsymbol{X}$ to the same $K-1$ dimensional space, i.e., $f : \mathbb{R}^p \to \mathbb{R}^{K-1}$, and use the angle between $f(\boldsymbol{X})$ and

$W_y$ for $y \in [K]$ to determine the prediction rule, i.e.,

$$\hat{y} = \underset{y}{\operatorname{argmin}} \angle(W_y, f(\boldsymbol{X})).$$

According to the "law of cosine", this is equivalent to

$$\hat{y} = \underset{y}{\operatorname{argmax}} \langle W_y, f(\boldsymbol{X}) \rangle, \tag{2.3}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product of two vectors. The inner product essentially plays the role of the functional margin in MAC, and the empirical risk minimization (ERM) is, therefore defined as follows:

$$\min_{f \in \mathcal{F}} \left\{ \sum_{i=1}^{n} l(\langle W_{y_i}, f(\boldsymbol{X}_i) \rangle) + \lambda J(f) \right\}, \tag{2.4}$$

where $l(\cdot)$ is the margin-based loss function defined by equation (2.1) and $J(f)$ denotes the penalty term with the tuning parameter $\lambda$, which controls the strength of regularization.

Considering the specialty of voxel-based neuroimaging classification, we narrow the function space $\mathcal{F}$ to linear functions, so that the coefficients of $f(\cdot)$ are voxel-wisely matched with the structure of the image covariate $\boldsymbol{X}$, i.e.,

$$\begin{aligned} f(\boldsymbol{X}) &= (f_1(\boldsymbol{X}), f_2(\boldsymbol{X}), \ldots, f_{K-1}(\boldsymbol{X}))^T, \\ f_j(\boldsymbol{X}) &= \beta_{j,0} + x_{1\beta_{j,1}} + \ldots + x_p \beta_{j,p} \text{ for } j \in [K-1]. \end{aligned} \tag{2.5}$$

Notice that $\boldsymbol{\beta}_j = (\beta_{j,1}, \ldots, \beta_{j,p})^T$ has a one-to-one correspondence with the imaging data $\boldsymbol{X} = (x_1, \ldots, x_p)^T$. Thus, it can be also defined in the original image space of the covariates. In this case, we denote $\boldsymbol{\beta}_j$ as the coefficient image of the fitted classifier.

For a $K$-category classification problem, we have $K-1$ coefficient images. In order to match the coefficient images with the $K$ class labels, we denote the reconstructed coefficient images $\boldsymbol{\beta}_y^*$, $y \in [K]$ of the same dimension of $\boldsymbol{\beta}_j$ as follows,

$$\boldsymbol{\beta}_y^* = \sum_{j=1}^{K-1} W_{y,j} \cdot \boldsymbol{\beta}_j, \tag{2.6}$$

where $W_{y,j}$ is the $j$-th element of the project class label $W_y$ in Equation (2.2) and "$\cdot$" denotes the element-wise product.

Note that $\boldsymbol{\beta}_y^*$ has the one-to-one correspondence with the class label $y$. Additionally, since $\sum_{y=1}^K W_y = 0$ according to Equation (2.2), we have the sum-to-zero constraint on $\boldsymbol{\beta}_y^*$'s as well, i.e., $\sum_{k=1}^K \boldsymbol{\beta}_y^* = 0$. These properties ensure that the reconstructed coefficient images are comparable with the coefficient images obtained from other linear classification models with the sum-to-zero constraint, such as logistic regression.

### 2.2.3 Spatial Smoothing Regularization

The penalty term $J(f)$ in problem (2.4) not only plays an important role of preventing the resulting classifier from over-fitting, but also helps to achieve some desired structure in the coefficient images. For image classification, unpenalized estimation often yields dense coefficients, but requires additional thresholding (or feature selection) to identify meaningful biomarkers. In contrast, the use of sparse penalties alone, such as lasso and the elastic net, leads to coefficient images with isolated voxels, which can be difficult to interpret. The use of spatial smoothing penalties not only captures the spatial smoothness in the image space, but also yields biologically interpretable coefficient images. For instance, Grosenick et al. (2013) proposed a spatial smoothing penalty, GraphNet, that incorporates the spatial structure in the elastic net penalization. However, the GraphNet penalty yields global smoothness in coefficient images, so it may be suboptimal in preserving sharp edges.

We introduce the generalized FL penalty (Tibshirani, 2011) to capture the spatial structure of imaging data. For an image $\boldsymbol{I} = \{\boldsymbol{I}(d) \in \mathbb{R} : d \in \mathcal{D}\}$, the discrete image intensities are evaluated at grid points $d = (d_1, d_2, d_3)^T \in R^3$ in a compact set $\mathcal{D}$. The FL penalty is a weighted mixture of the $L_1$ and TV penalty on the image intensities. The $L_1$ penalty encourages both shrinkage and sparseness (Tibshirani, 1996); whereas the TV penalty regularizes the differences between the consecutive elements in the estimation. We denote the latter as the TV-I penalty. Its discrete formulation is defined as follows:

$$\text{TV-I}(\boldsymbol{I}) = \sum_{d_1=1}^{D_1} \sum_{d_2=1}^{D_2} \sum_{d_3=1}^{D_3} ||\nabla \boldsymbol{I}_{d_1,d_2,d_3}||_1, \tag{2.7}$$

where $||\cdot||_1$ denotes the $L_1$ norm, $D_1$, $D_2$ and $D_3$ respectively represent the total number of voxels along each dimension, and $\nabla$ denotes the discrete differential operator such that $\nabla \boldsymbol{I}_{d_1,d_2,d_3} = (\nabla_1 \boldsymbol{I}_{d_1,d_2,d_3}, \nabla_2 \boldsymbol{I}_{d_1,d_2,d_3}, \nabla_3 \boldsymbol{I}_{d_1,d_2,d_3})^T$. Moreover, $\nabla_1 \boldsymbol{I}_{d_1,d_2,d_3}$ is defined as

$$\nabla_1 \boldsymbol{I}_{d_1,d_2,d_3} = \begin{cases} \boldsymbol{I}_{d_1,d_2,d_3} - \boldsymbol{I}_{d_1+1,d_2,d_3} & \text{if } 1 \leq d_1 \leq D_1 - 1, \\ 0 & \text{if } d_1 = D_1, \end{cases}$$

and $\nabla_2 \boldsymbol{I}_{d_1,d_2,d_3}$ and $\nabla_3 \boldsymbol{I}_{d_1,d_2,d_3}$ can be similarly defined.

The TV-I penalty penalizes the discrete gradient of the image function $\boldsymbol{I}(\cdot)$. It encourages the spatial smoothness of $I(\cdot)$, while capturing its sharp edges. This property allows us to efficiently detect important blobs. However, in some cases, the TV-I penalty tends to yield images with block-wise constant blobs (Rudin et al., 1992), which might erase too many details. For this reason, we introduce the second-order TV penalty, denoted TV-II, which can capture blobs with a continuous change of intensity by imposing the regularization on the Hessian matrix of $I(\cdot)$, which encourages the gradual fade of $I(\cdot)$ in the space. The discrete formulation of TV-II is defined as follows:

$$\text{TV-II}(\boldsymbol{I}) = \sum_{d_1=1}^{D_1-2} \sum_{d_2=1}^{D_2-2} \sum_{d_3=1}^{D_3-2} ||H(\boldsymbol{I}_{d_1,d_2,d_3})||_1, \tag{2.8}$$

where $H(I_{d_1,d_2,d_3}) = (\nabla_m(\nabla_{m'}(\boldsymbol{I}_{d_1,d_2,d_3})))_{1\leq m,m'\leq 3}$ and $||\cdot||_1$ denotes the entry-wise $L_1$ norm of a matrix.

Note that the calculation of both gradient and Hessian operators can be represented as matrix multiplication on the vectorized images. In particular, the TV-I($\boldsymbol{I}$) in (2.7) can be represented as

$$\text{TV-I}(\boldsymbol{I}) = ||\mathbf{D} \times \boldsymbol{I}||_1,$$

where $\mathbf{D}$ denotes the discrete derivative operator that contains the differencing operation along each of the 3 dimensions of the image domain.

Similarly, the TV-II penalty can be represented as

$$\text{TV-II}(\boldsymbol{I}) = ||\mathbf{D}_{II}\mathbf{D} \times I||_1,$$

where $\mathbf{D}_{II} = \text{diag}\{\mathbf{D}, \mathbf{D}, \mathbf{D}\}$ is a diagonal block matrix, with 3 copies of matrix $D$ representing the operations along each dimension.

For problem (2.4), we have $K-1$ coefficient images for a $K$ category classification problem and can denote $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{K-1})^T$ as the vector of all the image coefficients, as denoted in equation (2.5). The associated TV-I penalty is defined as

$$\text{TV-I}(\boldsymbol{\beta}) = \sum_{k=1}^{K-1} \text{TV-I}(\boldsymbol{\beta}_k) = \sum_{k=1}^{K-1} ||\mathbf{D} \times \boldsymbol{\beta}_k||_1 = ||\mathbf{C}_I \boldsymbol{\beta}||_1,$$

where $\mathbf{C}_I = [\mathbf{D}, \ldots, \mathbf{D}]$ is $K-1$ copies of the operator $\mathbf{D}$. Similarly, we can define

$$\text{TV-II}(\boldsymbol{\beta}) = ||\mathbf{C}_{II} \boldsymbol{\beta}||_1,$$

where $\mathbf{C}_{II} = [\mathbf{D}_{II}\mathbf{D}, \ldots, \mathbf{D}_{II}\mathbf{D}]$ is $K-1$ copies of the matrix $\mathbf{D}_{II}\mathbf{D}$.

Finally, the EMR problem in (2.4) can be reformulated as follows:

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^{(K-1)(p+1)}} \left\{ \sum_{i=1}^{n} l(\langle W_{y_i}, f(\boldsymbol{X}_i) \rangle) + \text{FL}(\boldsymbol{\beta}) \right\}, \tag{2.9}$$

where $l(\cdot)$ is the loss function in (2.1), $f(\cdot)$ is a system of linear functions defined in (2.5), and $\text{FL}(\boldsymbol{\beta}) = \lambda_1 ||\boldsymbol{\beta}||_1 + \lambda_2 ||\mathbf{C}\boldsymbol{\beta}||_1$ defines the FL penalty, in which $\lambda_1$ and $\lambda_2$ are two non-negative tuning parameters and $\mathbf{C} = \mathbf{C}_I$ for TV-I or $\mathbf{C}_{II}$ for TV-II.

### 2.2.4 Algorithm

The optimization in problem (2.9) is a mixture of smooth and non-smooth convex optimization. Many iterative proximal algorithms can be adopted here to solve this problem, such as ISTA and FISTA (Beck and Teboulle, 2009). However, the evaluation of the Lipschitz constant and the proximal operators can be computationally expensive in this case. Instead, we introduce an alternative direction method of multipliers (ADMM) (Boyd et al., 2011) algorithm to solve the optimization efficiently.

### 2.2.5 Alternative Direction Method of Multipliers

The ADMM algorithm (Boyd et al., 2011; Mota et al., 2011) was developed to handle large-scale convex optimization problems with the following separable and constrained structure:

$$\min_{\mathbb{X},\mathbb{Y}} \quad g_1(\mathbb{X}) + g_2(\mathbb{Y}) \quad \text{subject to} \quad \mathbb{A}_1\mathbb{X} + \mathbb{A}_2\mathbb{Y} = 0, \tag{2.10}$$

where $\mathbb{X} \in \mathbb{R}^p$ and $\mathbb{Y} \in \mathbb{R}^q$ are unknown parameters, $g_1(\mathbb{X})$ and $g_2(\mathbb{Y})$ are two closed convex functions, and $\mathbb{A}_1 \in \mathbb{R}^{m\times p}$ and $\mathbb{A}_2 \in \mathbb{R}^{m\times q}$ represent $m$ linear constraints on $\mathbb{X}$ and $\mathbb{Y}$, respectively. ADMM solves (2.10) by breaking them into smaller and simpler subproblems and solving them alternatively. Specifically, for the $t+1$ iteration,

$$\begin{aligned}
\mathbb{X}^{t+1} &= \operatorname*{argmin}_{\mathbb{X}} \left\{ g_1(\mathbb{X}) + \frac{\rho}{2} ||\mathbb{A}_1\mathbb{X} + \mathbb{A}_2\mathbb{Y}^t + \mathbf{u}^t||_2^2 \right\}, \\
\mathbb{Y}^{t+1} &= \operatorname*{argmin}_{\mathbb{Y}} \left\{ g_2(\mathbb{Y}) + \frac{\rho}{2} ||\mathbb{A}_1\mathbb{X}^{t+1} + \mathbb{A}_2\mathbb{Y} + \mathbf{u}^t||_2^2 \right\}, \\
\mathbf{u}^{t+1} &= \mathbb{A}_1\mathbb{X}^{t+1} + \mathbb{A}_2\mathbb{Y}^{t+1} + \mathbf{u}^t,
\end{aligned}$$

where $\rho$ denotes the augmented Lagrangian parameter, $\mathbf{u}$ is a vector of dual variables, and $||\cdot||_2$ denotes the $L_2$ Euclidean norm. The choice of $\rho$ affects the convergence rate of the algorithm (Boyd et al., 2011), and remains an open question in the literature. We implement our algorithm with $\rho = 1$, but it can be tuned in practice.

#### 2.2.5.1 Reformulation of ERM

We first reformulate the ERM (2.9) so that the ADMM algorithm can be applied smoothly. Note that the evaluation of the functional margins $\langle W_{y_i}, f(\boldsymbol{X}_i)\rangle$ consists of only linear operations. We construct a big matrix $\mathbf{A}$, such that the inner product can be simplified as one matrix multiplication, i.e.,

$$\langle W_{y_i}, f(\boldsymbol{X}_i)\rangle = \sum_{k=1}^{K-1} W_{y_i,k} \left(\langle \boldsymbol{X}_i, \boldsymbol{\beta}_k\rangle + \beta_{k,0}\right) = \mathbf{A}_{i,\cdot}\boldsymbol{\beta} \text{ for } i \in [n], \tag{2.11}$$

where $\mathbf{A}_{i,\cdot}$ denotes the $i$-th row of the matrix $\mathbf{A}$. The details for constructing such a matrix $\mathbf{A}$ can be found in Appendix A1.

The penalty term in (2.9) consists of a sum of two $L_1$ norms of vectors, and thus can be simplified as

$$||\mathbf{B}\boldsymbol{\beta}||_1 = \lambda_1||\boldsymbol{\beta}||_1 + \lambda_2||\mathbf{C}\boldsymbol{\beta}||_1,$$

where $\mathbf{B}^T = [\lambda_1\mathbf{I}, \lambda_2\mathbf{C}^T]$. With a little bit of adjustment to the notations, we use $\mathbf{I}$ to denote the identity matrix here. Furthermore, we reconstruct the differencing matrix $\mathbf{C}$ to a circulant matrix $\widetilde{\mathbf{C}}$ by adding some additional rows, and define $\widetilde{\mathbf{B}}^T = [\lambda_1\mathbf{I}, \lambda_2\widetilde{\mathbf{C}}^T]$ accordingly. Under this reformulation, the matrix $(\mathbf{I} + \widetilde{\mathbf{B}}^T\widetilde{\mathbf{B}})$ becomes a block circulant with a circulant block matrix and can be efficiently inverted by using the fast Fourier transform (FFT) (Chan et al., 1993).

For masked images, we introduce a recovering matrix $\mathbf{R}$ according to the masking matrix to recover the 3-D image structure with all the grid points in the space. A selection matrix $\mathbf{M}$ is then introduced to rule out the augmented rows added in $\widetilde{\mathbf{B}}$ and force the regions outside the mask to zeros. Therefore, we have

$$\mathrm{FL}(\boldsymbol{\beta}) = ||\mathbf{M}\widetilde{\mathbf{B}}\mathbf{R}\boldsymbol{\beta}||_1.$$

The EMR is then reformulated as

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^{(K-1)(p+1)}} \{\sum_{i=1}^{n} l((\mathbf{A}\boldsymbol{\beta})_i) + ||\mathbf{M}\widetilde{\mathbf{B}}\mathbf{R}\boldsymbol{\beta}||_1\}.$$

We further introduce some auxiliary constants and artificial variables to reformulate the problem in a desired form for the ADMM. This leads to our final ERM formulation as follows:

$$\begin{aligned} &\min_{\boldsymbol{\beta},\mathbf{v}_1,\mathbf{v}_2,\mathbf{v}_3} \quad \sum_{i=1}^{n} l(\mathbf{v}_{1i}) + ||\mathbf{M}\mathbf{v}_3||_1 \\ &\text{subject to} \quad \mathbf{v}_1 = \mathbf{A}\boldsymbol{\beta}, \quad \mathbf{v}_2 = \mathbf{R}\boldsymbol{\beta}, \quad \text{and} \quad \mathbf{v}_3 = \widetilde{\mathbf{B}}\mathbf{v}_2. \end{aligned} \tag{2.12}$$

Specifically, we set $\mathbb{X}^T = \left[\boldsymbol{\beta}^T, \mathbf{v}_3^T\right]$, $\mathbb{Y}^T = \left[\mathbf{v}_1^T, \mathbf{v}_2^T\right]$, $g_1(\mathbb{X}) = ||\mathbf{M}\mathbf{v}_3||_1$ and $g_2(\mathbb{Y}) = \sum_{i=1}^n l(\mathbf{v}_{1i})$., and denote

$$\mathbb{A}_1 = \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \text{ and } \mathbb{A}_2 = \begin{pmatrix} -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & -\widetilde{\mathbf{B}} \end{pmatrix},$$

and then the updating rules for the ADMM can be adopted smoothly for our problem.

#### 2.2.5.2 Closed-form solutions for the subproblems

We first demonstrate the solution of the optimization in the $\mathbb{X}$ block, which contains the following two subproblems:

$$\boldsymbol{\beta}^{t+1} = \arg\min_{\boldsymbol{\beta}} \left\{||\mathbf{A}\boldsymbol{\beta} - \mathbf{v}_1^t + \mathbf{u}_1^t||_2^2 + ||\mathbf{R}\boldsymbol{\beta} - \mathbf{v}_2^t + \mathbf{u}_2^t||_2^2\right\}, \tag{2.13}$$

$$\mathbf{v}_3^{t+1} = \arg\min_{\mathbf{v}_3} \left\{||\mathbf{M}\mathbf{v}_3||_1 + \frac{\rho}{2}||\mathbf{v}_3 - \widetilde{\mathbf{B}}\mathbf{v}_2^t + \mathbf{u}_3^t||_2^2\right\}. \tag{2.14}$$

**Solution for $\boldsymbol{\beta}$:**

The optimization of $\boldsymbol{\beta}$ in (2.13) is a quadratic minimization problem, which has a closed-form solution:

$$\boldsymbol{\beta}^{t+1} = \mathbf{K}^t - \mathbf{A}^T(\mathbf{I} - \mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A}\mathbf{K}^t = \mathbf{K}^t - \mathbf{H}_L\mathbf{H}_R^t, \tag{2.15}$$

where $\mathbf{K}^t = \mathbf{A}^T(\mathbf{v}_1^t - \mathbf{u}_1^t) + \mathbf{R}^T(\mathbf{v}_2^t - \mathbf{u}_2^t)$ and $\mathbf{H}_R^t = \mathbf{A}\mathbf{K}^t$. Moreover, $\mathbf{H}_L = \mathbf{A}^T(\mathbf{I} - \mathbf{A}\mathbf{A}^T)^{-1}$ is a fixed term across all iterations, so it can be precalculated.

**Solution for $\mathbf{v}_3$:**

Problem 2.14 can be solved by a proximal algorithm, the solution of which is given by

$$\mathbf{v}_3^{t+1} = \mathbf{M} \times \text{Soft}_{\rho^{-1}}\left(\widetilde{\mathbf{B}}\mathbf{v}_2^t - \mathbf{u}_3^t\right) - (\mathbf{I} - \mathbf{M})\left(\widetilde{\mathbf{B}}\mathbf{v}_2^t - \mathbf{u}_3^t\right), \tag{2.16}$$

where Soft$(\cdot)$ is a component-wide *soft thresholding* operator (Parikh and Boyd, 2013), denoted by $\mathbf{Soft}_\lambda(\mathbf{v}) = ((v_j - \lambda)_+ - (-v_j - \lambda)_+)_j$, in which $(x)_+ = \max\{x, 0\}$.

Next, we demonstrate the optimization of the $\mathbb{Y}$ block, which involves two variables $\mathbf{v}_1$ and $\mathbf{v}_2$. We apply the ADMM algorithms, and decompose it into the following two subproblems:

$$\mathbf{v}_1^{t+1} = \arg\min_{\mathbf{v}_1} \left\{\sum_{i=1}^{n} l(\mathbf{v}_{1i}) + \frac{\rho}{2}||\mathbf{A}\boldsymbol{\beta}^{t+1} - \mathbf{v}_1 + \mathbf{u}_1^t||_2^2\right\}, \tag{2.17}$$

$$\mathbf{v}_2^{t+1} = \arg\min_{\mathbf{v}_2} \left\{||\mathbf{R}\boldsymbol{\beta}^{t+1} - \mathbf{v}_2 + \mathbf{u}_2^t||_2^2 + ||\mathbf{v}_3^{t+1} - \widetilde{\mathbf{B}}\mathbf{v}_2 + \mathbf{u}_3^t||_2^2\right\}. \tag{2.18}$$

**Solution for $\mathbf{v}_1$:**

The optimization of $\mathbf{v}_1$ in (2.17) can be solved component-wisely by applying the Newton-Raphson method, i.e.,

$$\mathbf{v}_{1i}^{t+1} = \mathbf{v}_{1i}^{t} - \frac{l'(\mathbf{v}_{1i}^{t}) + \rho\left(\mathbf{v}_{1i}^{t} - \left(\mathbf{A}_{i\cdot}\boldsymbol{\beta}^{t+1} + \mathbf{u}_{1i}^{t}\right)\right)}{l''(\mathbf{v}_{1i}^{t}) + \rho}, \text{ for } i = [n], \tag{2.19}$$

where $l'(\cdot)$ and $l''(\cdot)$ are the first- and second-order derivatives of the loss function $l(\cdot)$, which are given as follows:

$$l'(u) = \begin{cases} -1 & \text{if } u < 0 \\ -e^{-u} & \text{if } u \geq 0 \end{cases} \text{ and } l''(u) = \begin{cases} 0 & \text{if } u < 0 \\ e^{-u} & \text{if } u \geq 0 \end{cases}.$$

To ensure convergence, we need to conduct multiple iterations in every Newton step. In our implementation, we only perform 1 iteration, which has been shown to result in sufficiently good convergence in practice.

**Solution for $\mathbf{v}_2$:**

The optimization of $\mathbf{v}_2$ in (2.18) is a standard quadratic programming problem, which has a closed-form solution:

$$\mathbf{v}_2^{t+1} = \left(\mathbf{I} + \widetilde{\mathbf{B}}^T\widetilde{\mathbf{B}}\right)^{-1}\left\{\left(\mathbf{R}\boldsymbol{\beta}^{t+1} + \mathbf{u}_2^t\right) + \widetilde{\mathbf{B}}^T\left(\mathbf{v}_3^{t+1} + \mathbf{u}_3^t\right)\right\}.$$

The direct inversion of the matrix $\mathbf{I}+\widetilde{\mathbf{B}}^T\widetilde{\mathbf{B}}$ may not be feasible due to the extra high dimensionality. We make use of its block circulant structure and solve the problem in $\mathbf{v}_2$ by FFT at a cost of $O(n \log n)$ operations (Afonso et al., 2010). Specifically, we have

$$\mathbf{v}_2^{t+1} = \mathit{ifft}\left(\mathit{fft}\left(\left(\mathbf{R}\boldsymbol{\beta}^{t+1} + \mathbf{u}_2^t\right) + \widetilde{\mathbf{B}}^T\left(\mathbf{v}_3^{t+1} + \mathbf{u}_3^t\right)\right) \div \mathit{fft}\left(\Gamma_1\right)\right), \tag{2.20}$$

where *fft* and *ifft* denote the 3-D FFT and inverse FFT operators, respectively, "$\div$" denotes the element-wise division, and $\Gamma_1$ is the first column of matrix $\mathbf{I} + \widetilde{\mathbf{B}}^T\widetilde{\mathbf{B}}$.

A complete ADMM updating procedure is summarized in algorithm 1. We list all the involved parameters in Table 2.1 for a convenient reference. The primal updates are discussed above. The dual updates are directly derived from the general updating rule of the ADMM algorithm. We conduct the primal and dual updates alternatively until the prespecified convergence criteria are satisfied. In particular, we check the relative change in the estimated coefficient $\beta$, and stop the algorithm if the total number of iterations exceeds a prespecified bound or the relative change is below a certain threshold, $\epsilon$, i.e.,

$$\frac{||\boldsymbol{\beta}^{t+1} - \boldsymbol{\beta}^{t}||}{||\boldsymbol{\beta}^{t}||} \leq \epsilon. \tag{2.21}$$

**Algorithm 1** ADMM algorithm for SMAC-I/II

```
Initialize primal variables $\boldsymbol{\beta}$, $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$ as $\mathbf{0}$.
Initialize dual variables $\mathbf{u}_1$, $\mathbf{u}_2$, $\mathbf{u}_3$ as $\mathbf{0}$.
Set $t = 0$, assign $\lambda_1, \lambda_2 \geq 0$.
Precompute $\mathbf{H}_L = \mathbf{A}^T \left(I - \mathbf{A}\mathbf{A}^T\right)^{-1}$
while $t \leq t_{max}$ do
  Primal update:
  $\boldsymbol{\beta}^{t+1} = \mathbf{K}^t - \mathbf{H}_L \mathbf{A} \mathbf{K}^t$    (2.15)
  $\mathbf{v}_3^{t+1} = \mathbf{M} \times \text{Soft}_{\frac{1}{\rho}} \left(\widetilde{\mathbf{B}}\mathbf{v}_2^t - \mathbf{u}_3^t\right) - (\mathbf{I} - \mathbf{M}) \left(\widetilde{\mathbf{B}}\mathbf{v}_2^t - \mathbf{u}_3^t\right)$    (2.16)
  $\mathbf{v}_{1i}^{t+1} = \mathbf{v}_{1i}^t - \frac{l'(\mathbf{v}_{1i}^t) + \rho\left(\mathbf{v}_{1i}^t - \left(\mathbf{A}_{i.}\boldsymbol{\beta}^{t+1} + \mathbf{u}_{1i}^t\right)\right)}{l''(\mathbf{v}_{1i}^t) + \rho}$, for $i = [n]$,    (2.19)
  $\mathbf{v}_2^{t+1} = \mathit{ifft}\left(\mathit{fft}\left(\left(\mathbf{R}\boldsymbol{\beta}^{t+1} + \mathbf{u}_2^t\right) + \widetilde{\mathbf{B}}^T \left(\mathbf{v}_3^{t+1} + \mathbf{u}_3^t\right)\right) \div \mathit{fft}\left(\Gamma_1\right)\right)$    (2.20)
  Dual update:
  $\mathbf{u}_1^{t+1} = \mathbf{A}\boldsymbol{\beta}^{t+1} - \mathbf{v}_1^{t+1} + \mathbf{u}_1^t$
  $\mathbf{u}_2^{t+1} = \mathbf{R}\boldsymbol{\beta}^{t+1} - \mathbf{v}_2^{t+1} + \mathbf{u}_2^t$
  $\mathbf{u}_3^{t+1} = \mathbf{v}_3 - \widetilde{\mathbf{B}}\mathbf{v}_2^{t+1} + \mathbf{u}_3^t$
  Convergence criteria:
  if $||\boldsymbol{\beta}^{t+1} - \boldsymbol{\beta}^t|| / ||\boldsymbol{\beta}^t|| > \epsilon$ then
    $t = t + 1$
  else
    break
    return $\boldsymbol{\beta} = \boldsymbol{\beta}^{t+1}$
  end if
end while
```

### 2.2.6 Simulation of synthetic data

To illustrate the finite sample performance of SMAC, we conducted simulation studies in both binary and multi-category cases.

#### 2.2.6.1 Generation of the synthetic data

In Simulation I, we simulated 2 classes of images of size $20 \times 20 \times 10$. The true signals for each class are denoted as $\theta_1$ and $\theta_2$ (see Figure 2.1), where $\theta_1$ has two ROIs and $\theta_2$ has three ROIs. The

**Table 2.1:** List of parameters in Algorithm 1.

| Parameter(s) | Description |
|---|---|
| $\boldsymbol{\beta}$ | Target variable in the optimization. |
| $\mathbf{v}_1$ | Auxiliary variable, $\mathbf{v}_1 = \mathbf{A}\boldsymbol{\beta}$. |
| $\mathbf{v}_2$ | Auxiliary variable, $\mathbf{v}_2 = \mathbf{R}\boldsymbol{\beta}$. |
| $\mathbf{v}_3$ | Auxiliary variable, $\mathbf{v}_3 = \widetilde{\mathbf{B}}\boldsymbol{\beta}$. |
| $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$ | Dual variables in the ADMM. |
| $\lambda_1, \lambda_2$ | Penalty strength for $L_1$ and TV-I/II respectively. |
| $\mathbf{A}$ | Matrix to compute functional margin, see equation (2.11). |
| $\mathbf{K}^t$ | Vector to be calculated for solving (2.15) ,$\mathbf{K}^t = \mathbf{A}^T(\mathbf{v}_1^t - \mathbf{u}_1^t) + \mathbf{R}^T(\mathbf{v}_2^t - \mathbf{u}_2^t)$. |
| $\widetilde{\mathbf{B}}$ | Augmented discrete operator for FL penalty, see Section 2.2.5.1 for details. |
| $\mathbf{M}$ | Selection matrix to rule out additional terms, see Section 2.2.5.1 for details. |
| $\mathbf{R}$ | Recovering matrix for masked images. |
| $\Gamma_1$ | The first column of matrix $\mathbf{I} + \widetilde{\mathbf{B}}^T\widetilde{\mathbf{B}}$. |

discriminating region between the two classes is the ROI represented by the region of the black triangular prism in the center, which contains 75 voxels in total. The image intensities in the three ROIs are 0, 1 and 2, respectively.

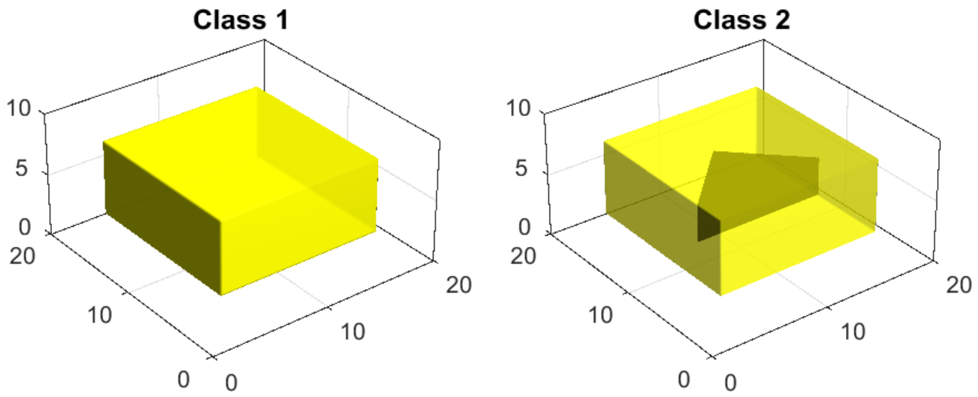


**Figure 2.1:** True signals for two classes of images in Simulation I. The left panel is the true image of class 1: the transparent and yellow regions represent the voxel values of 0 and 1, respectively. The right panel is the true image of class 2: the transparent, yellow and black regions represent $0, 1$ and 2, respectively.

In Simulation II, we considered classifying three classes of images. The image size is $32 \times 32 \times 4$, and the true signals are $\theta_1$, $\theta_2$ and $\theta_3$, which are graphically illustrated in Figure 2.2. The image intensities are 0 in the black regions and 1 in the white regions. The discriminating regions among the three classes located in the first and second diagonal blocks are marked in the red boxes.

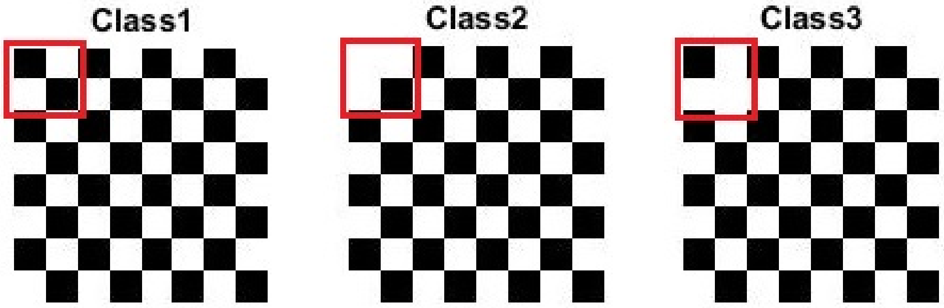


**Figure 2.2:** True signals for three classes of images in Simulation II. The three images are the top layer ($z = 1$) of the mean images for classes 1, 2, and 3, respectively. White represents the voxel value of 1, and black represents 0. The four layers ($z = 1, \dots, 4$) of the true image are identical within each class. The discriminating regions are marked in red boxes.

We generated noisy image samples by adding independent Gaussian noise at each voxel of the true signals, i.e., if the $i$-th image belongs to the $k$-th category, the associated noisy sample is given as

$$\boldsymbol{X}_i(t) = \boldsymbol{\theta}_k(t) + \epsilon_i(t) \text{ for all } t \in \mathcal{D} \text{ and } i \in [n], \tag{2.22}$$

where $\epsilon_i(t) \overset{iid}{\sim} N(0, \sigma^2)$ represents the Gaussian noise. For both simulation studies, we set $\sigma = 2$ for all samples.

### 2.2.7 Application: classification of MRI images from ADNI data

For the real data applications, we analyzed data from the ADNI study, a large-scale multi-site study that has collected MRI and PET images, CSF, and blood biomarkers, among other patient data. In AD, the most common form of dementia, the affected individual progressively develops disabilities in memory, language, and behavior, and the disease eventually results in death. A key goal of the ADNI study is to develop more sensitive and accurate biomarkers for the early detection of AD. The participants in the ADNI study include cognitively NCs, individuals with amnestic mild cognitive impairment (MCI), and subjects with AD. More information about this study can be found at the ADNI website (http://adni.loni.usc.edu/).

#### 2.2.7.1 Participants

In this chapter, we used a subset of baseline T1-weighted images from the ADNI study. After removing images with low quality, we obtained a dataset consisting of 749 samples (209 NC, 361 MCI and 179 AD). Table 2.2 summarizes the demographic information of all the subjects in our data analysis.

**Table 2.2:** Demographic information of all subjects in the ADNI data analysis. The unit for intracranial volume (ICV) is $1,000cm^3$. The means of age and ICV are reported, with standard deviations in parentheses.

| | Male | Female | Age | ICV |
|---|---|---|---|---|
| NC | 111 | 95 | 76.03 (4.95) | 1.27 (0.12) |
| MCI | 233 | 131 | 75.00 (7.38) | 1.29 (0.14) |
| AD | 98 | 81 | 75.50 (7.53) | 1.27 (0.15) |

#### 2.2.7.2 Image acquisition and processing

All images were preprocessed by a standard procedure (Guo et al., 2014), including anterior commissure and posterior commissure correction, N2 bias field correction, skull-stripping, intensity inhomogeneity correction, cerebellum removal, segmentation, and registration. We generated RAVENS-maps for the whole brain, using the deformation field obtained during registration (Davatzikos et al., 2001) and obtained 749 images of size 128 × 128 × 128. Considering that the variability of age, gender and whole-brain volume among different subjects may affect the classification results, we first removed those factors by fitting linear regression models at each voxel, and then built the classification model based on the residual images of these linear models.

## 2.3 Results

### 2.3.1 Comparison, tuning parameter selection and cross-validation

The proposed SMAC is designed to handle whole-brain volumetric data and detect disease-related regions without any prior spatial knowledge. To evaluate the performance of SMAC in these two tasks, we compared our method with other classifiers that can handle high-dimensional whole-brain volumetric data without any pre-screening procedure, and which also have the ability to yield volumetric coefficient images in the same space of the covariates. Under this guidance for comparison, we chose the following classifiers for neuroimaging classification: logistic regression using elastic-net regularization (EN-LR) (Casanova et al., 2011), logistic regression with the GraphNet penalty (GN-LR) (Grosenick et al., 2013) and SSVM with an FL penalty (Watanabe et al., 2014). Since SSVM was originally designed only for binary problems, we did not include it in the multi-category problems. To distinguish between SMAC with TV-I and TV-II penalties, we respectively denote them as SMAC-I and SMAC-II.

All the methods mentioned above involve two tuning parameters, $\lambda_1$ and $\lambda_2$. For consistent comparison, we denoted $\lambda_1$ as the tuning parameter of the sparse penalty terms for all methods. In SSVM, SMAC-I and SMAC-II, we denoted $\lambda_2$ as the tuning parameter of the total variation terms, whereas in EN-LR and GN-LR, we defined $\lambda_2$ as the parameter of the $L_2$ norm penalty. We conducted a grid search to select the best pair of the two parameters across a 21 by 21 log-based grid for the synthetic data, i.e., $\lambda_1 \otimes \lambda_2 \in \{0, 2^{-14}, 2^{-13}, \ldots, 2^5\}^{\otimes 2}$ and a smaller grid of $\lambda_1 \otimes \lambda_2 \in \{0, 2^{-13}, 2^{-11}, \ldots, 2^3, 2^5\}^{\otimes 2}$ for the real data.

For the analysis of the synthetic data, a data-rich scenario, we independently generated 30 training, 30 validation and 300 test samples for each class according to (2.22), which yielded 60 training, 60 validation and 600 test samples in Simulation I and 90 training, 90 validation and 900 test samples in Simulation II. We used the training samples to build models for each combination of $\lambda_1$ and $\lambda_2$, and evaluated the models on the validation samples to calculate the tuning classification accuracy and area under the curve (AUC) in the associated receiver operating characteristic (ROC) analysis. Based on the validation results, we picked the models with the highest classification accuracy. If ties occurred, we chose the models with highest AUC among them. If we still obtained multiple models, the one with a larger spatial penalty ($\lambda_2$) was selected as our final model. We applied the final model to the test samples to evaluate the classification performance. To validate the stability of the methods, we repeated the experiments for 50 iterations, and reported the means and standard deviations of the results.

For the real data analysis, we applied a stratified sampling on the whole dataset and split it into training (60%), validation (20%) and test (20%) sets, so that the proportions of NC, MCI and AD subjects were similar across the different sets. We used a validation and evaluation procedure that was similar to what we used in the simulation study. We repeated the above random split 30 times and recorded the means and standard deviations of the results.

### 2.3.2 Results from synthetic data analysis

#### 2.3.2.1 Cross-validation and tuning results

The mean validation accuracy matrices from 50 iterations of the simulation studies are given in Figure 2.3. In Simulation I (binary case), EN-LR yielded lower validation accuracy for most of the sparse estimation, i.e., $\lambda_1 \in \{2^1, \ldots, 2^5\}$. SSVM yielded higher tuning accuracies for the sparse and patched estimation, i.e., $\lambda_1 \otimes \lambda_2 \in \{0, 2^{-14}, \ldots, 2^{-8}\} \otimes \{2^{-5}, \ldots, 2^{-3}\}$. GN-LR achieved very good validation accuracy when the sparsity and smoothness levels were relatively high, but yielded low accuracy when the sparsity level was too high, i.e., $\lambda_1 \in \{2^4, 2^5\}$. SMAC-I and SMAC-II achieved overall higher validation accuracy and were more sensitive to the change in tuning parameters. In particular, the SMAC methods were more sensitive to the penalty level of the total variation than the sparse term. This is mainly explained by the spatial smoothness assumption of the imaging data.

The results of Simulation II are similar to those of Simulation I. The sparse method EN-LR yielded low validation accuracy for most combinations of the tuning parameters. GN-LR and SMAC achieved high accuracy under a relatively high sparsity level and a moderate smoothness penalty level, i.e., $\lambda_1 \otimes \lambda_2 \in \{2^{-5}, 2^{-4}, 2^{-3}\}^{\otimes 2}$.

#### 2.3.2.2 Receiver operating characteristic (ROC) analysis and classification accuracy

The ROC analysis can simultaneously evaluate the true positive rate and the false positive rate for a binary classifier under different thresholds. The AUC numerically measures the performance of a classifier in the ROC analysis. When dealing with the multi-category cases, the ROC analysis can be implemented using the "one vs. the rest" strategy, i.e., transforming it into multiple binary problems. We conducted the ROC analysis for both binary and multi-category problems, randomly picked one result from the 50 iterations, and plotted the associated ROC curves; see Figures 2.4 and 2.5. The numerical results for all iterations are summarized in Tables 2.3 and 2.4.

In the binary classification example, SMAC-I achieved the highest classification accuracy of 96.52% and the largest AUC of 99.58%, followed by an accuracy of 96.17% and an AUC of 99.39%

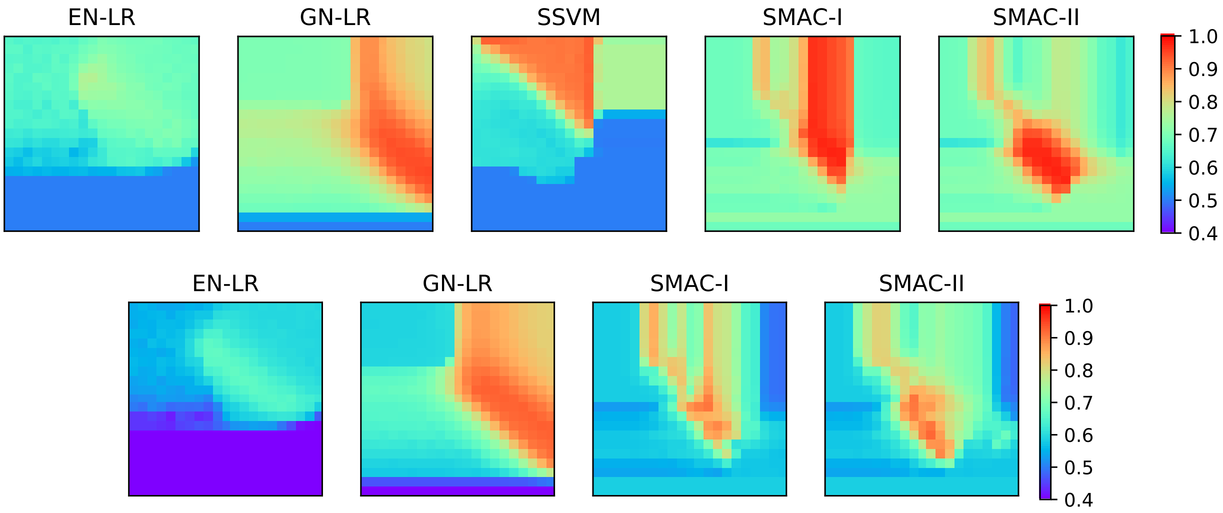


**Figure 2.3:** Validation accuracies for synthetic studies. The top row of 5 panels (from left to right) respectively correspond to the validation accuracy matrices of EN-LR, GN-LR, SSVM, SMAC-I and SMAC-II for the binary synthetic data. The bottom row of 4 panels (from left to right) respectively correspond to the validation accuracy matrices of EN-LR, GN-LR, SMAC-I and SMAC-II for the multi-category synthetic data. Each entry of the matrix is the tuning accuracy for the corresponding combination of $\lambda_1$ and $\lambda_2$. The vertical direction of the matrix represents the value of $\lambda_1$, from top to bottom being $\{0, 2^{-14}, 2^{-13}, \ldots, 2^5\}$, and the horizontal direction represents $\lambda_2$, from left to right being $\{0, 2^{-14}, 2^{-13}, \ldots, 2^5\}$.

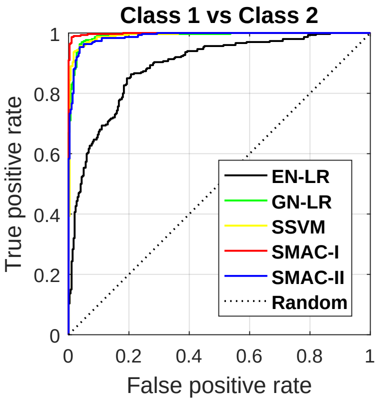


**Figure 2.4:** Receiver operating characteristic (ROC) analysis for the binary synthetic data based on 600 test samples.

from SMAC-II. GN-LR and SSVM yielded accuracies of 93.04% and 92.23%, and AUC values of 98.11% and 97.91%, respectively. EN-LR achieved an accuracy of 73.75% and an AUC of 81.70%.

In the multi-category cases, SMAC-I and SMAC-II yielded the highest respective accuracies of 94.70% and 94.19%, as well as the largest AUC values in all three classes. GN-LR achieved an accuracy of 92.17%. EN-LR yielded an accuracy of 64.89% and had the smallest AUC values in all three classes.

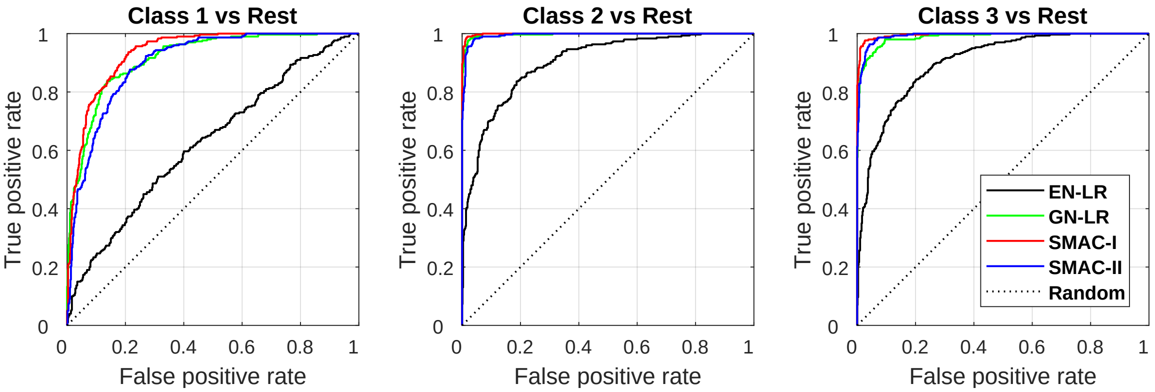


**Figure 2.5:** Receiver operating characteristic (ROC) analysis for the multi-category synthetic data based on 900 test samples. Each panel represents the ROC curves evaluated using the "one-versus-the-rest" strategy.

**Table 2.3:** Comparison of the classification results of binary synthetic data. Classification accuracy (ACC), true positive rate (TPR), true negative rate (TNR) and area under the ROC curve (AUC) are presented as percentages. Means from 50 iterations are reported, with standard deviations in parentheses.

| Method | ACC | TPR | TNR | AUC |
|---|---|---|---|---|
| EN-LR | 73.75(5.24) | 76.44(7.00) | 71.05(8.50) | 81.70(5.99) |
| GN-LR | 93.04(1.75) | 93.01(1.94) | 93.07(1.92) | 98.11(0.79) |
| SSVM | 92.23(2.06) | 94.11(2.88) | 90.35(3.37) | 97.91(1.00) |
| SMAC-I | **96.52**(1.21) | **95.83**(2.16) | **97.21**(1.30) | **99.51**(0.31) |
| SMAC-II | 96.17(1.32) | 95.54(1.98) | 96.79(1.72) | 99.39(0.36) |

**Table 2.4:** Comparison of the classification results of multi-category synthetic data. Classification accuracy (ACC) and area under the ROC curve (AUC1-3) for classes 1, 2 and 3 are presented as percentages. Means from 50 iterations are reported, with standard deviations in parentheses.

| Method | ACC | AUC1 | AUC2 | AUC3 |
|---|---|---|---|---|
| EN-LR | 64.89(3.29) | 63.96(2.70) | 86.41(3.02) | 86.54(3.22) |
| GN-LR | 92.17(1.13) | 90.74(4.60) | 99.55(0.16) | 99.23(0.26) |
| SMAC-I | **94.70**(1.10) | 95.65(1.42) | **99.85**(0.06) | **99.66**(0.13) |
| SMAC-II | 94.19(0.86) | **95.85**(1.12) | 99.71(0.07) | 99.64(0.10) |

In both simulation studies, the spatial methods were more stable in terms of the classification results and yielded smaller standard deviations among the 50 iterations. The sparse method EN-LR delivered sparse estimation consisting of isolated voxels, and thus yielded unstable models. In particular, its variable selection results varied a lot in different iterations.

#### 2.3.2.3 Visualization and interpretation of coefficient images

We plotted all the coefficient images to illustrate the estimation and identification of those critical regions for classifying the samples. The plot of the coefficient images in Simulation I (see Figure 2.6) reveals that EN-LR yielded a sparse coefficient image, consisting of isolated voxels; whereas all the other spatial penalized methods produced smooth coefficient images, clearly indicating the triangular discriminating region in the center. SSVM and SMAC-I both yielded clear boundaries between the predictive and irrelevant regions. SSVM contained many false positive voxels in the background; whereas SMAC-I had a "clean" background. GN-LR yielded smooth coefficient images with blurred boundaries around the triangular region and also contained some false positives in the background. SMAC-II yielded a similarly smooth coefficient image with many fewer false positives.

In the multi-category example, we illustrated the reconstructed coefficient images (defined in Equation 2.6) from SMAC-I/II and compared them with the penalized logistic regression methods (EN-LR and GN-LR). Since the three coefficients for each method summed to zero, we only displayed first two of them, i.e. $\hat{\boldsymbol{\beta}}_1$ and $\hat{\boldsymbol{\beta}}_2$. The estimated coefficient images obtained from EN-LR consist of isolated voxels. GN-LR and SMAC-II yielded smooth patched estimations, but with a blurred boundary. The coefficient images from SMAC-I clearly captured the first and second diagonal block regions of the checkerboard image, which are the most critical regions for discriminating the three classes.

Accurately capturing the key discriminating regions is a requirement of a good image classifier. In both simulation studies, the sparsity-only classifier EN-LR underperformed due to the ignorance of the spatial structure. GN-LR and SMAC-II tended to yield smooth critical regions in which the image intensities continuously varied across voxels. SSVM and SMAC-I were able to capture the critical regions with clear boundaries. SMAC-I and SMAC-II achieved fewer false positives in the irrelevant regions, while the other methods contained either isolated or patchy false positives in the background. For these particular synthetic data, SMAC-I delivered the most competitive performance. This was mainly due to the assumption of patchy constant patterns in the discriminating regions. SMAC-II may have potential advantages when those regions have continuously varying intensities.

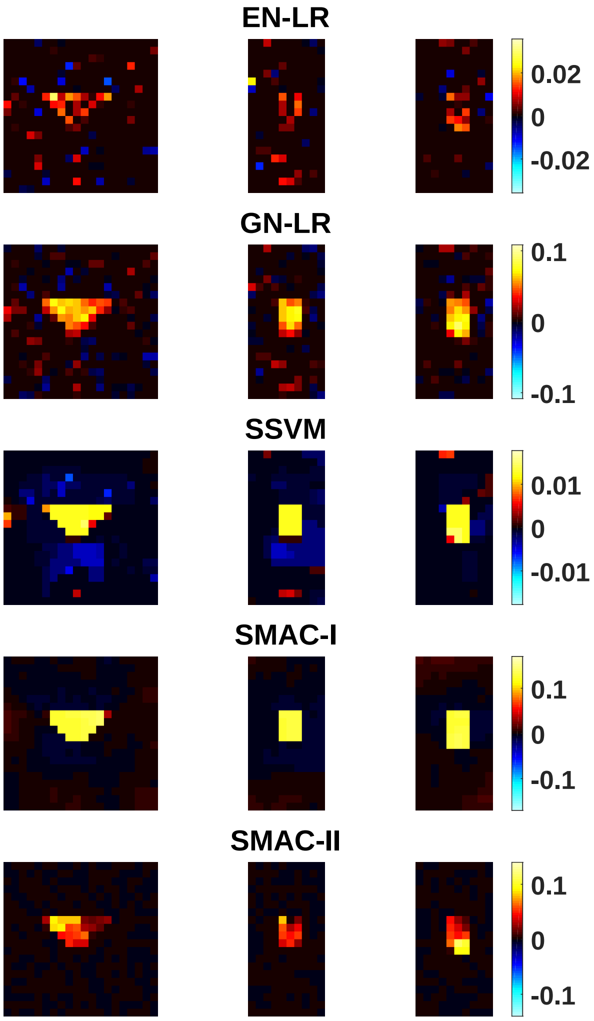


**Figure 2.6:** Estimated coefficient images obtained from five classification methods in Simulation I. The 5 panels display the coefficient images of EN-LR, SSVM, GN-LR, SMAC-I and SMAC-II. Each coefficient image is displayed in three respective directions: transverse, coronal and sagittal, from left to right. The center of all the images is located at $(10, 10, 5)$.

#### 2.3.2.4 Model sensitivity on training sample size and noise level

To further analyze the stability of the proposed methods, we conducted a comprehensive sensitivity analysis on the sample size and noise level. In particular, the sample size analysis was done by repeating the experiment in Simulation I with the training sample size ranging from 10 to 100. The validation and test sample sizes were not changed, and the noise level remained the same for different sample sizes, i.e., $\epsilon_i(t) \overset{iid}{\sim} N(0, 4)$. A similar model selection procedure as that used in Simulation I was used, and we report the test results in Table 2.5 and Figure 2.8. The noise level sensitivity analysis was done by fixing the training sample size ($n = 30$) and varying the standard

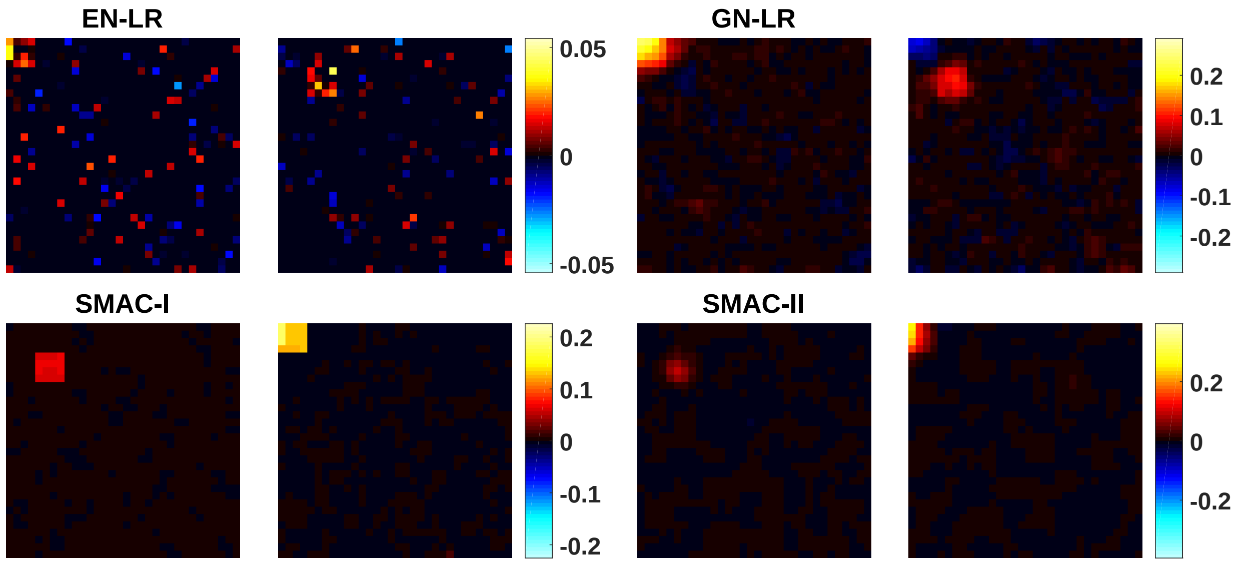


**Figure 2.7:** Estimated coefficient images obtained from four classification methods in Simulation II.The top panels are the respective coefficients from EN-LR and GN-LR, and the bottom panels are the coefficients from SMAC-I and SMAC-II. The first two coefficient images ($\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$) of each classifier are displayed. The coefficients from SMAC-I and SMAC-II are obtained using Equation (2.6). All the coefficient images are displayed in the transverse direction, centered at $(16, 16, 1)$.

deviation of the noise added to each voxel from $\sigma = 1$ to 4. The test results are summarized in Table 2.6 and Figure 2.9.

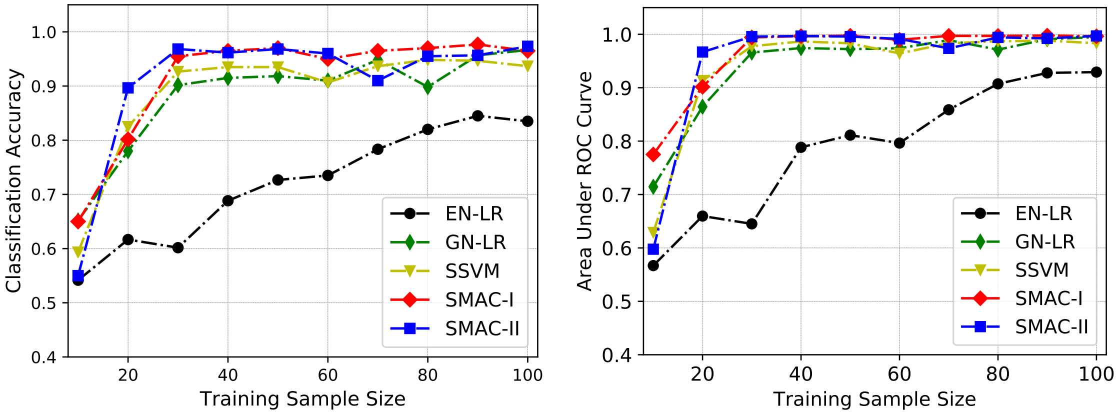


**Figure 2.8:** Classification results under different training sample sizes. The left panel displays the classification accuracy and right panel displays the area under the ROC curve.

From the sensitivity analysis, we conclude that the proposed SMAC methods can achieve high accuracies and AUCs with very limited training samples, e.g., $n \leq 50$, and yield very competitive performance in the cases of mid-noise levels, e.g., $\sigma \in (2, 3)$.

**Table 2.5:** Sample size sensitivity analysis. Columns represent different training sample sizes. Classification accuracy (ACC) and area under the ROC curve (AUC) are presented as percentages. The evaluation is based on 600 test samples.

| Method | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | **ACC** | | | | | |
| EN-LR | 54.17 | 61.67 | 60.17 | 68.83 | 72.67 | 73.50 | 78.33 | 82.00 | 84.50 | 83.50 |
| GN-LR | 65.17 | 78.00 | 90.17 | 91.50 | 91.83 | 91.00 | 94.83 | 89.83 | 95.67 | 96.67 |
| SSVM | 59.33 | 82.50 | 92.67 | 93.50 | 93.50 | 90.67 | 93.67 | 94.83 | 94.67 | 93.67 |
| SMAC-I | 65.00 | 80.17 | 95.50 | 96.50 | 97.00 | 95.00 | 96.50 | 97.00 | 97.67 | 96.50 |
| SMAC-II | 55.00 | 89.67 | 96.83 | 96.17 | 96.83 | 96.00 | 91.00 | 95.50 | 95.67 | 97.33 |
| | | | | | **AUC** | | | | | |
| EN-LR | 56.69 | 65.97 | 64.51 | 78.84 | 81.12 | 79.63 | 85.88 | 90.72 | 92.77 | 92.92 |
| GN-LR | 71.45 | 86.44 | 96.59 | 97.43 | 97.23 | 97.39 | 98.91 | 97.13 | 99.09 | 99.48 |
| SSVM | 62.89 | 91.34 | 97.77 | 98.66 | 98.32 | 96.42 | 98.33 | 98.70 | 98.81 | 98.33 |
| SMAC-I | 77.50 | 90.15 | 99.42 | 99.65 | 99.73 | 98.98 | 99.71 | 99.69 | 99.76 | 99.63 |
| SMAC-II | 59.76 | 96.68 | 99.57 | 99.67 | 99.58 | 99.14 | 97.37 | 99.42 | 99.24 | 99.69 |

**Table 2.6:** Noise level sensitivity analysis. Columns represent different standard deviations of noise. Classification accuracy (ACC) and area under the ROC curve (AUC) are presented as percentages. The evaluation is based on 600 test samples.

| Method | 1.0 | 1.25 | 1.5 | 1.75 | 2.0 | 2.25 | 2.5 | 2.75 | 3.0 | 3.25 | 3.5 | 3.75 | 4.0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | **ACC** | | | | | | |
| EN-LR | 97.00 | 94.83 | 84.33 | 78.50 | 73.50 | 72.50 | 64.50 | 64.17 | 58.67 | 60.67 | 54.17 | 58.17 | 57.17 |
| GN-LR | 100.00 | 99.00 | 98.67 | 97.17 | 91.00 | 88.00 | 81.00 | 77.50 | 72.00 | 69.17 | 66.67 | 63.17 | 61.83 |
| SSVM | 99.67 | 98.83 | 96.50 | 93.50 | 90.67 | 84.83 | 80.50 | 76.33 | 72.50 | 68.83 | 65.33 | 57.00 | 56.00 |
| SMAC-I | 98.50 | 98.83 | 98.17 | 96.17 | 95.00 | 93.50 | 90.83 | 88.00 | 83.00 | 71.67 | 67.50 | 65.83 | 58.00 |
| SMAC-II | 99.33 | 98.67 | 97.00 | 98.00 | 96.00 | 94.33 | 89.33 | 80.83 | 76.17 | 82.00 | 69.17 | 69.33 | 70.83 |
| | | | | | | | **AUC** | | | | | | |
| EN-LR | 99.97 | 98.81 | 92.90 | 87.89 | 79.63 | 80.64 | 69.03 | 67.56 | 61.13 | 63.61 | 54.55 | 60.72 | 58.42 |
| GN-LR | 100.00 | 99.96 | 99.91 | 99.65 | 97.39 | 95.42 | 89.43 | 85.46 | 80.61 | 76.38 | 73.31 | 69.15 | 67.66 |
| SSVM | 99.98 | 99.97 | 99.59 | 98.20 | 96.42 | 92.83 | 88.57 | 83.96 | 79.15 | 74.78 | 70.52 | 59.82 | 58.46 |
| SMAC-I | 99.84 | 99.94 | 99.87 | 99.31 | 98.98 | 97.86 | 96.75 | 95.19 | 91.30 | 78.25 | 74.06 | 73.44 | 62.56 |
| SMAC-II | 100.00 | 100.00 | 99.82 | 99.87 | 99.14 | 98.69 | 95.37 | 88.76 | 85.01 | 90.72 | 77.93 | 78.52 | 78.13 |

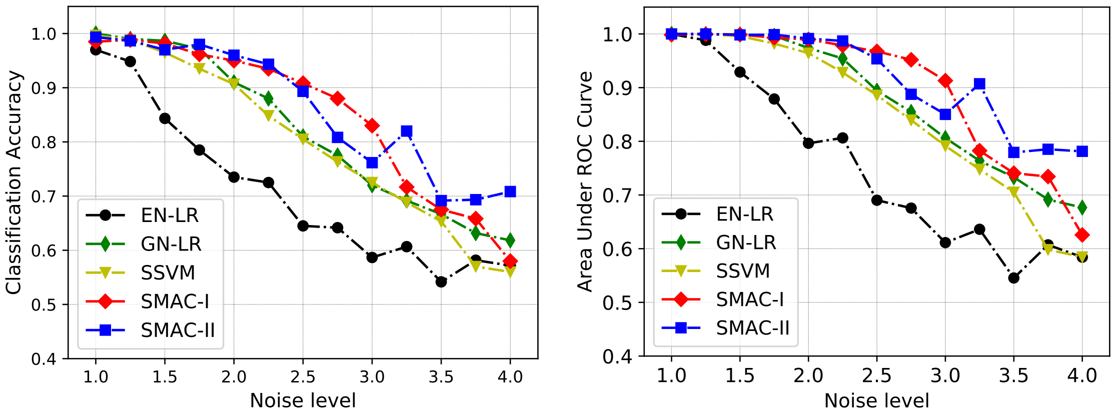


**Figure 2.9:** Classification results under different noise levels. The left panel displays classification accuracy and the right panel displays the area under the ROC curve.

### 2.3.3 Results from ADNI data

We conducted both binary and multi-category classification experiments using the ADNI data. In particular, we classified all possible pairs of the three classes as binary problems (NC vs AD, NC vs MCI and MCI vs AD) and identified AD, MCI and NC simultaneously as a three-category problem. The classification accuracies are presented in Tables 2.7 and 2.8. After we obtained the best tuning parameters from the 30 iterations of the three-way split, we refitted the model using all the data with the selected parameters and registered the coefficient images to the Montreal Neurological Institute (MNI)-152 template (Fonov et al., 2011). A plot of these coefficients in the orthogonal views is provided in Figures 2.10 and 2.11.

#### 2.3.3.1 ROC analysis and classification accuracy

In the classification problem of NC vs AD, SMAC-I and SMAC-II achieved the highest two accuracies of 89.12% and 88.33% respectively. The other three methods yielded similar accuracies between 86% and 87%. In the classification of MCI vs AD and NC vs MCI, the overall accuracies were lower. This may be partially explained by the uncertainty involved in the cognitive test for identifying MCI and the heterogeneity within the MCI group. EN-LR and SMAC-I/II yielded accuracy values that were very close in these tasks. SSVM was outperformed by the other methods, and could not capture informative signals in the classification of NC vs MCI. Notice that GN-LR achieved the highest values of AUC in all three binary classification problems. This is explained by the merit of the logistic loss in terms of estimating the “soft” class label, i.e. the associated probability (Liu et al., 2011). SMAC-I/II also yielded very competitive AUC values (second best in all three problems).

**Table 2.7:** Comparison of the binary classification results of MRI Data. Classification accuracy (ACC) and area under the ROC curve (AUC) are presented as percentages. Means from 30 iterations are reported, with standard deviations in parentheses.

| | **NC vs AD** | | **MCI vs AD** | | **NC vs MCI** | |
|---|---|---|---|---|---|---|
| Method | ACC | AUC | ACC | AUC | ACC | AUC |
| EN-LR | 86.84(3.13) | 93.33(1.90) | **69.22**(2.65) | 66.31(4.90) | 70.35(2.07) | 73.33(2.41) |
| GN-LR | 86.23(2.78) | **95.95**(1.36) | 65.67(3.23) | **68.15**(5.39) | 68.55(2.29) | **76.69**(2.99) |
| SSVM | 86.14(3.41) | 92.65(3.02) | 48.07(5.47) | 56.35(4.21) | 63.72(0.00) | 50.00(0.00) |
| SMAC-I | **89.12**(2.30) | 93.92(1.20) | 69.13(2.72) | 67.24(4.62) | **70.68**(2.43) | 75.31(2.22) |
| SMAC-II | 88.33(3.32) | 93.97(1.44) | 69.19(3.59) | 66.86(4.64) | 70.38(2.51) | 76.35(2.48) |

For the simultaneous classification of NC, MCI and AD, the classification accuracies are lower than those for the binary cases. SMAC-I/II yielded higher accuracies (53.22% and 52.68%) compared to those achieved by EN-LR and GN-LR (49.32% and 49.75%). The AUC values for MCI (AUC2) were lower than the ones for NC and AD, which was consistent with the results in the binary cases. SMAC-II achieved the best and second best values for AUC1 and AUC3, indicating a better detection rate for NC and AD.

**Table 2.8:** Comparison of the 3-category classification results of MRI data. Classification accuracy (ACC) and area under the ROC curve (AUC1-3) with respective reference labels NC, MCI and AD are presented as percentages. Means from 30 iterations are reported, with standard deviations in parentheses.

| Method | ACC | AUC1 | AUC2 | AUC3 |
|---|---|---|---|---|
| EN-LR | 49.32(3.18) | 72.39(3.15) | 50.68(4.66) | 77.56(4.44) |
| GN-LR | 49.75(2.55) | 77.90(2.99) | **51.94**(4.83) | **82.89**(1.45) |
| SMAC-I | **53.22**(2.90) | 81.01(3.23) | 50.31(5.03) | 77.53(3.63) |
| SMAC-II | 52.68(4.20) | **81.75**(3.85) | 48.85(4.79) | 78.21(3.67) |

#### 2.3.3.2 Clinically meaningful coefficient images

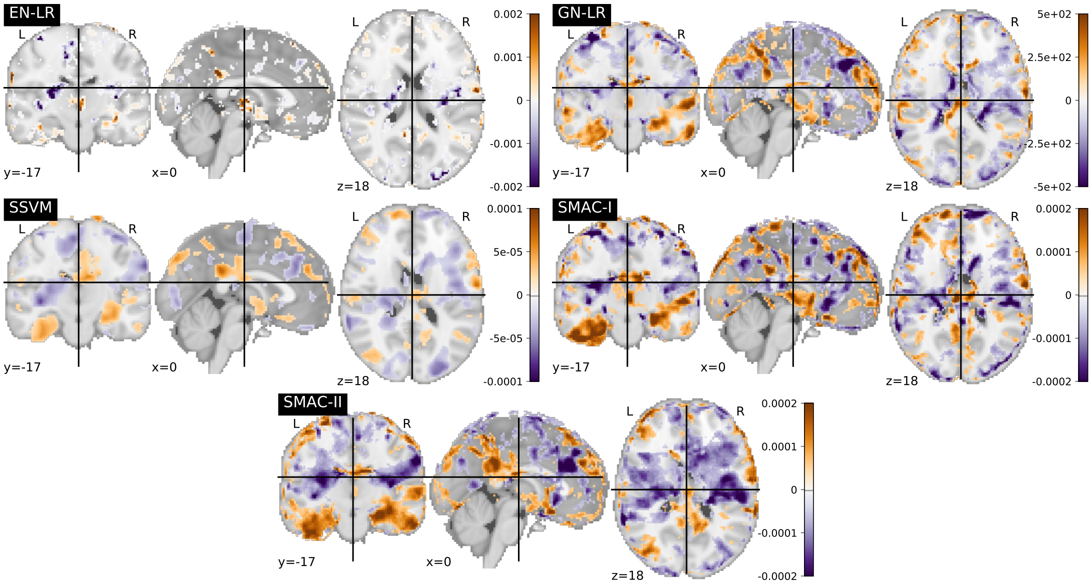


**Figure 2.10:** Estimated coefficient images obtained from five classification methods in the binary ADNI study. The five plots are the respective coefficient images from EN-LR, GN-LR, SSVM, SMAC-I and SMAC-II. Each coefficient is displayed in the views of coronal, sagittal and transverse planes. The slices are located at $(0, -17, 18)$.

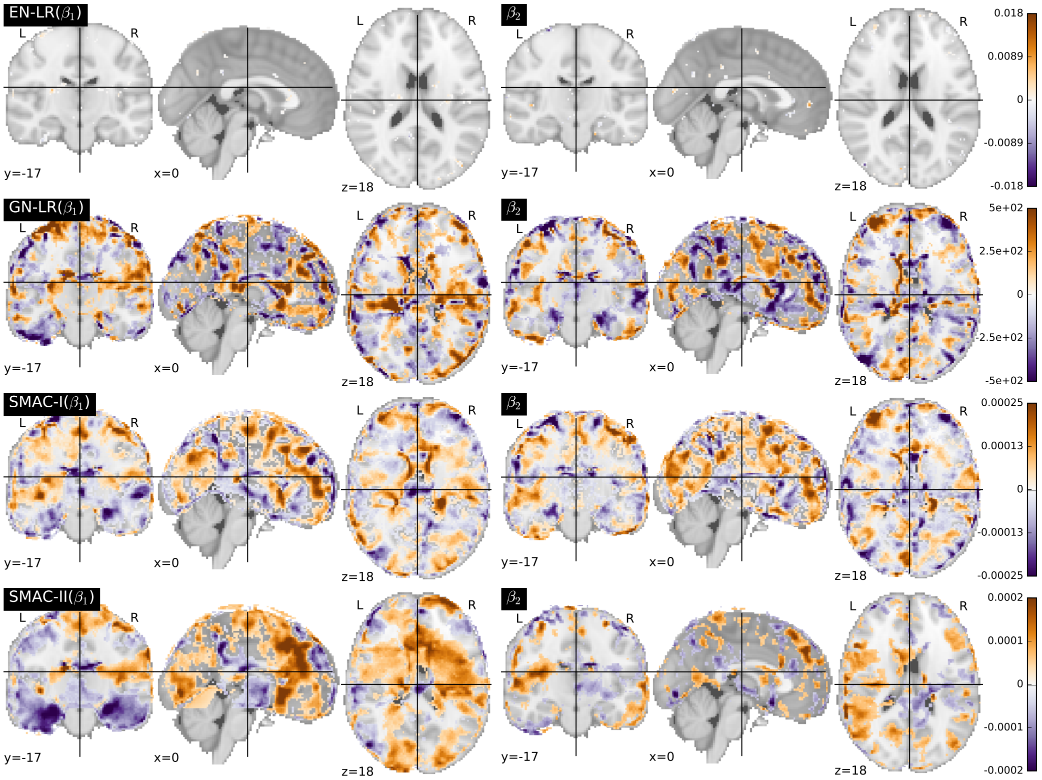


**Figure 2.11:** Estimated coefficient images obtained from four classification methods in the multi-category ADNI study. The four rows of plots are the respective coefficient images from EN-LR, GN-LR, SMAC-I and SMAC-II. The first two coefficient images ($\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$) of each classifier are displayed. The coefficients from SMAC-I and SMAC-II are obtained using Equation (2.6). Each coefficient is displayed in the views of coronal, sagittal and transverse planes. The slices are located at $(0, -17, 18)$.

Different from our synthetic imaging data, the MRI images of human brains are much more complex. Due to heterogeneity across subjects and the potential bias in the registration process, the boundaries between the discriminating regions and the background may not be as sharp as they are in the synthetic data. The patchy patterns may not be a perfect assumption for this case, but still help the classifiers to recover the predictive regional signals. From the plots in Figures 2.10 and 2.11, we can clearly see consistent patterns across the coefficient images from different spatial methods. The sparse method EN-LR delivers ultra-sparse estimation that is difficult to interpret biologically. Notice that, among all the spatial methods, SMAC-II can recover more smooth and patchy signals, while screening out the irrelevant regions in the brain. This will make it easier to identify ROIs in the coefficient images of SMAC-II.

Comparing Figures 2.10 and 2.11, we can see that for each spatial method, the effective regions of $\boldsymbol{\beta}$ in Figure 2.10 and the first coefficient $\boldsymbol{\beta}_1$ in Figure 2.11 are relatively consistent, but the intensity values have the opposite signs, i.e., the regional effects are opposite. This is mainly because the positive class label in the binary problem is AD while in the multi-category problem, class label 1 is associated with NC.

By overlaying the coefficient images from SMAC-I/II on the MNI-152 ROI template, we are able to identify several significant discriminating regions, such as the frontal gyrus, hippocampus, and right fornix. Many papers in the existing literature have shown that these regions are potentially related to the development of MCI and AD. For instance, the hippocampal region is involved in memory processes that deteriorate with the development of AD. The structure of the hippocampus is altered by the degenerative processes associated with AD, and loss of the hippocampal volume occurs at a rate that is approximately two to four times faster in patients with AD than in age-matched healthy controls (West et al., 1994; Dubois et al., 2014).

### 2.3.4 Computational considerations

In the MATLAB implementation of our algorithms, most of the computation is realized through matrix operations. For moderate image sizes (e.g., total number of voxels less than $10^4$), our methods converge very fast compared to the others. For ultra-high-dimensional imaging data (e.g., total number of voxels greater than $10^6$), the matrix operations require more memory usage. Furthermore, for all the classifiers used in our comparison, the regularization parameters significantly affect the convergence and computational cost of the algorithms. We ran all the programs on the same type of computer (Intel Xeon E5-2643 v3 @ 3.40GHz) with the same random-access memory (8 GB DDR3 at 1600 MHz). All algorithms were set with the same maximum number of iterations ($t_{max} = 1500$) and convergence threshold ($\epsilon = 5 \times 10^{-5}$) as defined in (2.21). We plotted the mean computational time from all 5 classifiers among 50 iterations in Simulation I (see Figure 2.12). EN-LR required the shortest time for this classification problem. The variation in the computational time was very small. This was mainly due to the simplicity of the EN-LR model. SMAC-I required the second shortest computational time, followed by SMAC-II. This was because the second-order total variation involved computation of the discrete Hessian operators, which had larger sizes than the gradient operators in SMAC-I. GN-LR also yielded very competitive computational speed. SSVM was out-performed by the other classifiers in terms of the computational speed. This was mainly due to the splitting scheme in its ADMM algorithm and the heavy computational load in optimization involving the non-smoothing hinge loss.

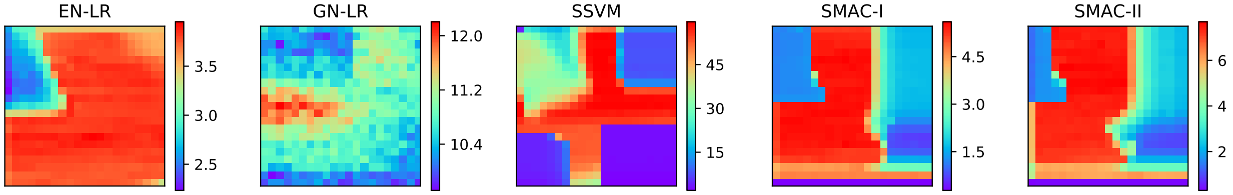


**Figure 2.12:** Mean computational time for each method in Simulation I. In each plot, the vertical direction represents the value of $\lambda_1$, from top to bottom being $\{0, 2^{-14}, 2^{-13}, \ldots, 2^5\}$, and the horizontal direction corresponds to $\lambda_2$, from left to right being $\{0, 2^{-14}, 2^{-13}, \ldots, 2^5\}$.

## 2.4 Discussion

In this chapter, we propose a SMAC for neuroimaging classification. Our method achieves the desired spatial sparsity and smoothness in the coefficient images via imposing the FL penalty. It improves the accuracy in both binary and multi-category classification problems. Both the simulation studies and the real data application demonstrate the usefulness of the proposed method.

Numerous classification studies in the literature have used the ADNI data, but their data collection and evaluation procedures may vary significantly. A direct comparison of the results may not be a reasonable way to evaluate the methods. For example, Dukart et al. (2011) achieved 100% accuracy on the classification of NC vs AD, while we obtained 89.12% accuracy for the same problem. However, their study assessed only 13 NC and 21 AD subjects; whereas our study assessed the 749 participants in the ADNI study. Moreover, they used pre-computed ROI statistics from both MRI and fluorodeoxyglucose-PET images as predictors; whereas we directly classified the baseline MRI data and automatically extracted the regional information during the estimation procedure. An advantage of our proposed method is that we can handle imaging data with limited pre-processing, and still produce reasonably good results. This can be valuable when the prior knowledge of spatial segmentation is not available.

We introduce an efficient algorithm using ADMM to solve the corresponding large-scale optimization problem in our method. Specifically, we propose a novel splitting scheme in ADMM and reduce the complexity of optimization. As a result, our algorithm performs more efficiently than the ADMM algorithms in the existing literature, such as Ye and Xie (2011) and Watanabe et al. (2014). Moreover, the proposed algorithm is very flexible and can be applied to solve various other prediction problems within the *loss + penalty* framework. We have included the implementation of the squared error loss in our package, which allows users to perform spatial regularized high-dimensional regression analysis. Details about this extension are included in Appendix A2.

One potential limitation of the proposed method is the underlying assumption of the spatially clustered patterns in the true coefficient images. This is a reasonable assumption in most neuroimaging applications. However, if the overall predictive effect is scattered around most of the image space, this method can be inefficient due to the complexity of the optimization problem.

There are several possible interesting extensions of the proposed method for future exploration. For example, all linear classifiers are built based on the assumption that all images are perfectly aligned and the predictive regions are consistent across all subjects within the same class. These assumptions can be violated in practice, both due to the non-negligible registration error and the heterogeneous structures within the population. The estimation and predictive performance can be strongly impacted by this issue. One possible future research direction is to handle this heterogeneity problem.

## Appendix

### A1. The construction of matrix A

To simplify the inner product in the loss function, we need to construct a big matrix $A$ to summarize all the linear operations. First, we denote

$$W_{y_i} = (W_{y_i,1}, W_{y_i,2}, \ldots, W_{y_i,K-1})^T \in \mathbb{R}^{K-1}$$

and construct a matrix $\widetilde{W}_Y$ consisting of a stack of diagonal matrices, i.e.,

$$\widetilde{W}_Y = \left[ \begin{array}{c} W_{Y,1}, W_{Y,2}, \cdots, W_{Y,K-1} \end{array} \right],$$

where $W_{Y,l} = \mathrm{diag}\{W_{y_1,l}, \ldots, W_{y_n,l}\}$ for $l = 1, \ldots, K-1$. Moreover, we denote $\widetilde{\boldsymbol{X}} = \mathrm{diag}\{\boldsymbol{X}, \cdots, \boldsymbol{X}\}$ as a matrix consisting of $K-1$ copies of the original covariate matrix on the diagonal. In particular, the columns of 1's are added at the first column of the covariate matrix to include the intercepts in the computation. Then, we define $A = \widetilde{W_Y}\widetilde{\boldsymbol{X}}$, and it can be verified that

$$A\boldsymbol{\beta} = \begin{bmatrix} \langle W_{y_1}, f(\boldsymbol{X}_1) \rangle \\ \vdots \\ \langle W_{y_n}, f(\boldsymbol{X}_n) \rangle \end{bmatrix}.$$

## A2. Spatial regularized regression

The ADMM algorithm proposed in this chapter is quite flexible and can be extended to solve other problems. In this section, we introduce the extension of our algorithm to solve a spatial regularized regression problem.

For the regression problem, the response variable $y_i$ can be a continuous measure of a certain clinical index. Denote the covariate image as $\boldsymbol{X}_i$. The regularized regression problem is given by

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\mathrm{argmin}} \left\{ \frac{1}{2} \sum_{i=1}^{n} (y_i - \boldsymbol{X}_i \boldsymbol{\beta})^2 + FL(\boldsymbol{\beta}) \right\}.$$

This is analogous to equation (2.9), by letting $K = 2$ and adopting the square loss, i.e., $l(u) = u1^2$. Both the first- and second-order total variations can be applied here.

We adopt the reformulation of equation (2.9) and construct a similarly constrained optimization to problem (2.12), i.e.,

$$\min_{\boldsymbol{\beta}, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3} \quad \frac{1}{2} \sum_{i=1}^{n} (y_i - \mathbf{v}_{1i})^2 + ||\mathbf{M}\mathbf{v}_3||_1$$

$$\text{subject to} \quad \mathbf{v}_1 = \boldsymbol{X}\boldsymbol{\beta}, \quad \mathbf{v}_2 = \mathbf{R}\boldsymbol{\beta}, \quad \text{and} \quad \mathbf{v}_3 = \widetilde{\mathbf{B}}\mathbf{v}_2.$$

Here, all the variable are the same as those in the algorithm for SMAC, but with fixed $K = 2$. The only change is that the loss function part becomes the squared error loss and $\mathbf{A}$ in problem (2.12) becomes $\boldsymbol{X}$. Thus, the solutions for $\boldsymbol{\beta}$, $\mathbf{v}_2$ and $\mathbf{v}_3$ remain the same by setting $\mathbf{A} = \boldsymbol{X}$ in (2.15).

The subproblem involving $\mathbf{v}_1$ becomes the following:

$$\mathbf{v}_1^{t+1} = \arg\min_{\mathbf{v}_1} \left\{ \frac{1}{2} ||\mathbf{Y} - \mathbf{v}_1||_2^2 + \frac{\rho}{2} ||A\boldsymbol{\beta}^{t+1} - \mathbf{v}_1 + \mathbf{u}_1^t||_2^2 \right\},$$

where $\mathbf{Y} = [y_1, \ldots, y_n]^T$. This is a standard quadratic programming problem for which the solution is given by

$$\mathbf{v}_1^{t+1} = \frac{1}{1+\rho} \left( \mathbf{Y} + \mathbf{A}\boldsymbol{\beta}^{t+1} - \mathbf{u}_1^t \right). \tag{2.23}$$

Therefore, the whole algorithm can be summarized as algorithm 2.

**Algorithm 2** ADMM algorithm for spatial regularized regression

```
**Initialize** primal variables $\boldsymbol{\beta}$, $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$ as $\mathbf{0}$.
**Initialize** dual variables $\mathbf{u}_1$, $\mathbf{u}_2$, $\mathbf{u}_3$ as $\mathbf{0}$.
**Set** $t = 0$, assign $\lambda_1, \lambda_2 \geq 0$.
**Set** $\mathbf{A} = \mathbf{X}$ and $K = 2$.
Precompute $\mathbf{H}_L = \mathbf{A}^T \left( I - \mathbf{A}\mathbf{A}^T \right)^{-1}$
**while** $t \leq t_{max}$ **do**
    *Primal update:*
    $\boldsymbol{\beta}^{t+1} = \mathbf{K}^t - \mathbf{H}_L \mathbf{A} \mathbf{K}^t$    (2.15)
    $\mathbf{v}_3^{t+1} = \mathbf{M} \times \text{Soft}_{\frac{1}{\rho}} \left( \widetilde{\mathbf{B}}\mathbf{v}_2^t - \mathbf{u}_3^t \right) - (\mathbf{I} - \mathbf{M}) \left( \widetilde{\mathbf{B}}\mathbf{v}_2^t - \mathbf{u}_3^t \right)$    (2.16)
    $\mathbf{v}_1^{t+1} = \frac{1}{1+\rho} \left( \mathbf{Y} + \mathbf{A}\boldsymbol{\beta}^{t+1} - \mathbf{u}_1^t \right)$    (2.23)
    $\mathbf{v}_2^{t+1} = \mathit{ifft} \left( \mathit{fft} \left( \left( \mathbf{R}\boldsymbol{\beta}^{t+1} + \mathbf{u}_2^t \right) + \widetilde{\mathbf{B}}^T \left( \mathbf{v}_3^{t+1} + \mathbf{u}_3^t \right) \right) \div \mathit{fft} \left( \Gamma_1 \right) \right)$    (2.20)
    *Dual update:*
    $\mathbf{u}_1^{t+1} = \mathbf{A}\boldsymbol{\beta}^{t+1} - \mathbf{v}_1^{t+1} + \mathbf{u}_1^t$
    $\mathbf{u}_2^{t+1} = \mathbf{R}\boldsymbol{\beta}^{t+1} - \mathbf{v}_2^{t+1} + \mathbf{u}_2^t$
    $\mathbf{u}_3^{t+1} = \mathbf{v}_3 - \widetilde{\mathbf{B}}\mathbf{v}_2^{t+1} + \mathbf{u}_3^t$
    *Convergence criteria:*
    **if** $||\boldsymbol{\beta}^{t+1} - \boldsymbol{\beta}^t|| / ||\boldsymbol{\beta}^t|| > \epsilon$ **then**
        $t = t + 1$
    **else**
        **break**
        **return** $\boldsymbol{\beta} = \boldsymbol{\beta}^{t+1}$
    **end if**
**end while**
```

# CHAPTER 3
# SVSIR: Subject Variant Scalar-on-Image Regression

## 3.1 Introduction

The use of imaging biomarkers to predict clinical outcomes is of great impact in public health. Many studies have demonstrated that medical images deliver clinically important information, which has been widely used to explore the pathophysiology of certain diseases and assist diagnosis and treatments (Giedd et al., 1999; Ogawa et al., 1990; Khoo et al., 1997).

Numerous statistical and machine learning tools are developed to analyze medical images. One branch of these methods use voxel-wise or mass-univariate regressions to explore the relationship between medical images and certain clinical measurements that are important for the diagnosis of the diseases (Ashburner and Friston, 2000; Karnath et al., 2004). In this approach, one regresses voxel-wise image measurements on the clinical scores, and builds regression models at each voxel. This procedure is denoted as a image-on-scalar regression. Some statistics, such as the p-value of the t-test for the regression coefficients are computed at each voxel to construct a statistical parametric map (Friston et al., 1994). Such maps can be used to extract the imaging biomarkers that are highly correlated with the corresponding clinical measurements(Smith et al., 2006). Despite of its advantages, the usefulness of image-on-scalar regression in prediction is limited. Multivariate or "decoding" models are widely used to overcome this limitation (Haynes and Rees, 2006; Haxby et al., 2001; Norman et al., 2006). Instead of fitting regression models at each voxel, these methods use the entire images as the covariates to predict the scalar response, such as the clinical measurements. We denote this approach as the scalar-on-image regression model. Usually, the linear relationship between the response and the covariate images are presumed, so that the coefficients of the model lie on the same space as the covariate image do. We refer such coefficients as the coefficient images and they are of great importance both in the prediction of clinical outcomes and in the identification of pathologically relevant imaging biomarkers.

The scalar-on-image regression can be viewed as a special case of the functional linear regressions (FLR) (Ramsay and Silverman, 2005; Ferraty and Vieu, 2006) or high-dimensional linear models (HDM) (Bühlmann and Van De Geer, 2011), depending on how the imaging data is addressed in the model. Correspondingly, there are two major approaches to estimate the scalar-on-image regression models. In one approach, each voxel value of the imaging data are treated as a feature in the model. The clinical scores are predicted through a multi-variate regression:

$$\begin{aligned} y_i &= \beta_0 + \langle \boldsymbol{X}_i, \boldsymbol{\beta} \rangle + \epsilon_i \\ &= \beta_0 + \sum_{j=1}^{p} \boldsymbol{X}_i(t_j)\boldsymbol{\beta}(t_j) + \epsilon_i \end{aligned}$$

where $\beta_0$ represents the intercept term, the operation $\langle \cdot, \cdot \rangle$ denotes the inner product of the covariate and the coefficient image, and $t_j$ represents the location index of j-th voxel. The dimension of such multiple linear regression model is defined as the total number of voxels in the covariate images, i.e., $p = |\mathcal{D}|$. This number can be very huge for the medical images. For example, to analysis a typical T1 MRI images of size $256 \times 256 \times 256$ using this method, one will need to build a $16,777,216$ dimensional regression model. Moreover, due to the cost of image acquisitions, the sample size $n$ is usually very small, compared to the model dimension. It makes the scalar-on-image regression a very ill-posed problem (Hadamard, 1902). Many regularized methods are developed to address this issue. For example, Carroll et al. (2009) proposed a regression model with Elastic-Net penalty (Zou and Hastie, 2005) to analyze the functional MRI data, and Toiviainen et al. (2014) applied the Lasso regression (Tibshirani, 1996) to identify the brain regions that respond to musical stimuli. These sparse regularization methods successfully solve the issue caused by the high dimensionality and are able detect image features that are potentially related to the responses. However, their coefficient images are lack of spatial smoothness, thus are not helpful to extract clinically interpretable imaging biomarkers. Recently, the spatial regularization methods are getting popular due to their well-structured coefficient images and improved prediction performance. For instance, (Grosenick et al., 2013) proposed the GraphNet penalty as a generalized Elastic-Net penalty that incorporates local graphical structures; Liu et al. (2018) introduced a spatial regularization framework using first and second order total variation penalties for both classification and regression problems. The spatial regularizations utilize the locally spatial structure of the imaging data and yield coefficient images with more spatial smoothness.

The other approach to estimate the scalar-on-image regression is using the functional linear regressions (FLR) techniques. Instead of fitting an ultra high dimensional model, these methods treat the images as functional data over the image domain, and construct the coefficient images using certain basis functions. For example, Reiss and Ogden (2007) proposed the functional principal component regression (FPCR) and functional partial least squares (FPLS) approaches to estimate the scalar-on-image regression, and Reiss et al. (2015) applied the wavelet basis to construct spatially smooth regression coefficients. These methods can achieve spatially smoothing coefficient images if the basis functions are chosen properly. We will focus on the functional techniques to solve the scalar-on-image regression problem in this chapter.

Another area of study of this chapter is the heterogeneity of the imaging data. In the current literature, most of the scalar-on-image regression models assume the homogeneous regression relationship and apply a unified model for all the subjects. However, this assumption may not hold in practice. Although the biological structures of human organs are relatively consistent across the whole population, certain diseases can cause structural damages in different regions for different patients, thus yield heterogeneous patterns. For example, Figure 3.1 displays the brain images of two head & neck patients. The tumors are marked by the red circles and located in different brain regions. The homogeneous models will underperform both in term of estimation and prediction in this case. Many mixture regression models are proposed to address the heterogeneity issue. Examples include but are not limited to Viele and Tong (2002), Hurn et al. (2003), Hoshikawa (2013), and Wang et al. (2016). However, their estimation can be unstable when the number of the mixture components is large; and their prediction accuracies may not be improved comparing to the homogeneous model. The prediction rule of mixture regression models is an weighted average of the mixing components based on the estimated posterior probabilities, thus ignores the heterogeneity and causes biased prediction (Hoshikawa, 2013).

The main goal of this chapter is to develop a Subject Variant Scalar-on-Image Regression (SVSIR) model for the heterogeneous imaging data. Three major methodological contributions are summarized as follows:

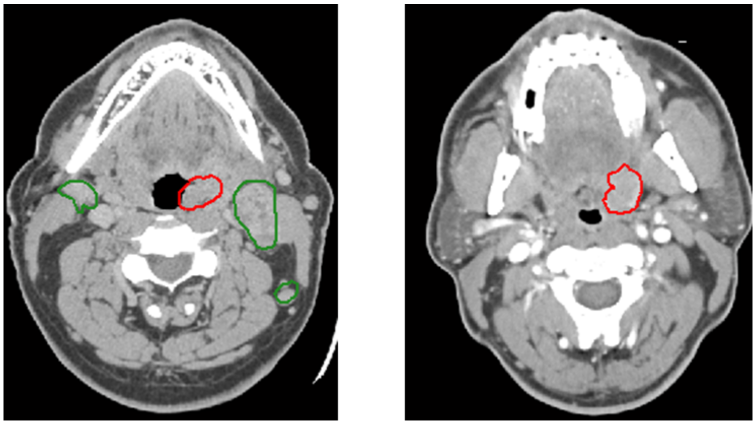


**Figure 3.1:** The brain images of two head & neck patients. The tumors are marked in the red circles.

- The proposed SVSIR model effectively utilizes spatial structure of the imaging data, and achieves clinically meaningful coefficient images with sparse and smooth signals;
- We introduce an adaptive algorithm that solve the estimation of model coefficients efficiently;
- The heterogeneity of imaging data is addressed and the prediction performance is improved.

The rest of this chapter is organized as follows. In Section 3.2, we introduce the SVSIR modeling framework and its key components. In Section 3.3, we establish the algorithms for the estimation procedure, model selection, and the prediction. Section 3.4 demonstrates the theoretical properties of the proposed method. We use extensive simulation experiments to exam the performance of SVSIR in various settings in Section 3.5. The real application in the ADNI study is provided in Section 3.6. Section 3.7 summarizes the chapter with discussion.

## 3.2 Methods and models

In this section, we introduce the technical details of the proposed SVSIR including the homogeneous and heterogeneous coefficients and the Potts prior that incorporate the spatial sparsity of the image data.

### 3.2.1 Data structure and the homogeneous models

Let $\boldsymbol{X}(t) \in R$, $\forall t \in \mathcal{D}$ represent the image intensity, where $\mathcal{D}$ denotes the image domain, which can be a bounded 2-D surface or 3-D volume, and $t$ is the corresponding location index, which can be a vector of length 2 or 3 respectively, according to the dimension of image domain $\mathcal{D}$. Using these notations, a scalar-on-image regression model is given as follows,

$$y_i = \beta_0 + \int_{t \in \mathcal{D}} \boldsymbol{X}_i(t)\boldsymbol{\beta}(t)dt + \epsilon_i; \qquad \text{for } i = 1, \ldots, n, \tag{3.1}$$

where $y_i \in \mathbb{R}$ represents the scalar responses, $\boldsymbol{X}_i$ denotes the covariate image, which is a real valued function over the domain $\mathcal{D}$, $\boldsymbol{\beta}$ is the coefficient image corresponding to $\boldsymbol{X}_i$, which is referred as disease map in this chapter, $\beta_0 \in \mathbb{R}$ denotes the intercept term and $\epsilon_i$ represents the independent noise. Model (3.1) is very general and it covers many existing methods in the literature. Examples include but are not limited to (Shen and Zhu, 2015), (Kang et al., 2016) and (Liu and Yan, 2017).

The framework in (3.1) is referred as the homogeneous model in this chapter. It assumes a homogeneous regression relationship between the responses $y_i$'s and the covariates $\boldsymbol{X}_i$'s, and all the subjects share a common coefficient image $\boldsymbol{\beta}$. In practice, this assumption may be violated by the heterogeneities among the samples. For example, in neuroimaging studies, the patients belonging to different subtypes of a certain disease may have different pathological patterns thus should not share one common disease maps. In particular, the location of affected regions in medical images may vary across different subjects, and the number of such regions may be more than one and different from one subject to another. In that case, the modeling framework in (3.1) may fail or under-perform due to such heterogeneities across subjects.

### 3.2.2 Subject-specific models

Subject-specific models are potential candidates for solving the heterogeneity issue. Using the same notation for model (3.1), a subject-specific linear regression model is given by

$$y_i = \beta_0 + \int_{t \in \mathcal{D}} \boldsymbol{X}_i(t)\boldsymbol{\beta}_i(t)dt + \epsilon_i; \qquad \text{for } i = 1, \ldots, n. \tag{3.2}$$

Note that the only difference between (3.1) and (3.2) is that we use a subject-specific coefficient image $\boldsymbol{\beta}_i$ instead of a common $\boldsymbol{\beta}$ for each sample. One one hand, this model allows the samples to perform their unique regression relationship thus is quite flexible. On the other hand, despite the flexibility, there are two major limitations of such models. First, model (3.2) is not identifiable, i.e., we cannot construct $n$ coefficient images using just $n$ samples. Another limitation is the ignorance of the population structure, which makes the model lacking interpretability and prediction power.

In most neuroimaging studies, the heterogeneous disease patterns often exists in the form of sub-groups or sub-clusters. The disease maps may be similar within each group and different across groups. Thus, it is reasonable to assume that there exists a set of coefficient images, $\mathcal{B} = \{\boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\beta}^{(M)}\}$, where $M << n$ is the total number of sub-groups, that covers most of the pathological patterns of the disease. Therefore, instead of a using distinct $\boldsymbol{\beta}_i$ for each subject, we may assume $\boldsymbol{\beta}_i = \boldsymbol{\beta}^{(m)}$ if subject $i$ belongs to the $m$-th group. This assumption naturally leads to the consideration of a mixture regression model (Viele and Tong, 2002; Hurn et al., 2003; Hoshikawa, 2013; Wang et al., 2016), i.e.

$$y_i = \sum_{m=1}^{M} \tau_{im} \left[ \int_{t \in \mathcal{D}} \boldsymbol{X}_i(t) \boldsymbol{\beta}^{(m)}(t) dt \right] + \epsilon_i; \qquad \text{for } i = 1, \ldots, n, \tag{3.3}$$

where $\tau_{im}$ is the posterior probability of subject $i$ belonging to the group $m$, i.e.,

$$\tau_{im} = P(\boldsymbol{\beta}_i = \boldsymbol{\beta}^{(m)} | \boldsymbol{X}_i, y_i).$$

Compared with the homogeneous model (3.1) and the subject-specific model (3.2), the mixture regression model (3.3) has a good balance in terms of flexibility. It allows different disease maps for different subgroups while still being identifiable and estimable.

Although model (3.3) serves as a good compromise, it may underperform when the total number of the mixture components is large. Moreover, there are still no clear guideline to select the number of mixture components in the literature. People usually use cross validation or Bayesian information criteria (BIC) to determine this number, but the performance is not stable. Furthermore, the prediction of mixture regression depends on the estimation of the posterior probability for each group. Such estimation requires strong assumptions on the joint distribution of the covariates and the response (Hoshikawa, 2013).

We introduce a hidden binary masking image $\boldsymbol{B}_i$ to characterize the heterogeneity and propose a novel Subject Variant Scalar-on-Image Regression (SVSIR) model, which is formulated as follows,

$$y_i = \beta_0 + \int_{t\in\mathcal{D}} \boldsymbol{X}_i(t)\boldsymbol{B}_i(t)\boldsymbol{\beta}(t)dt + \epsilon_i; \quad \text{for } i = 1, \ldots, n. \tag{3.4}$$

This model (3.4) is motivated by considering both population structure and the individual heterogeneity. At the population-level, the intrinsic biological structures (e.g. brain structure) in the medical images are relatively consistent across different people. This inspires us to use a homogeneous disease map $\boldsymbol{\beta}$ that characterizes the common regression relationship between the clinical responses and medical images. At the individual-level, patients may present different affected regions in the images. We introduce the subject-specific masking image $\boldsymbol{B}_i$'s to capture this heterogeneity. On the one hand, they generate subject-specific disease patterns from the homogeneous disease map, i.e.,

$$\boldsymbol{\beta}_i(t) = \boldsymbol{B}_i(t)\boldsymbol{\beta}(t), \quad \forall t \in \mathcal{D}.$$

On the other hand, they play a role of subject-specific feature extraction procedure that selects different regions in the images as subject-specific covariates, i.e., $\tilde{\boldsymbol{X}}_i = \boldsymbol{X}_i \circ \boldsymbol{B}_i$. The overlap of all the $B_i$'s characterizes the overall sparsity of the disease map $\boldsymbol{\beta}$, and is denoted as the population masking image $\boldsymbol{B}$, i.e.,

$$\boldsymbol{B}(t) = \begin{cases} 1 & \text{if one of } \boldsymbol{B}_i(t) = 1 \\ 0 & \text{if all of } \boldsymbol{B}_i(t) = 0. \end{cases} \tag{3.5}$$

#### 3.2.2.1 Homogeneous disease map and its Potts prior

The population-level disease map $\boldsymbol{\beta}$ in the model (3.4) critically determines the regression relationship between the clinical responses and the medical images, and should reveal the pathological patterns in the images. The biological structure of the human organs naturally yields locally smooth patterns in the medical images, and the disease-related patterns usually appear at certain locations instead of random regions. Due to these characteristics of the neuroimagings, the disease map should achieve both spatial smoothness and sparsity, and be able to detect the possible disease-related regions while ruling out the irrelevant ones, such as background or normal tissues.

To incorporate spatial smoothness, we assume that the coefficient image $\boldsymbol{\beta}$ lies on the reproducing kernel Hilbert space ($\mathcal{RKHS}$) with a radial based kernel. In this chapter, we mainly focus on the Gaussian kernel, which is given as follows,

$$K(s,t) = \exp\left\{-\frac{\|s-t\|^2}{\sigma^2}\right\}, \tag{3.6}$$

where $s, t \in \mathcal{D}$ denote two different location indices in the image domain and $\sigma$ represents the bandwidth of the kernel function. The space $\mathcal{H}$ induced by this kernel is a space of smooth functions over the image domain $\mathcal{D}$. The smoothness of the functions is controlled by the band-width $\sigma$.

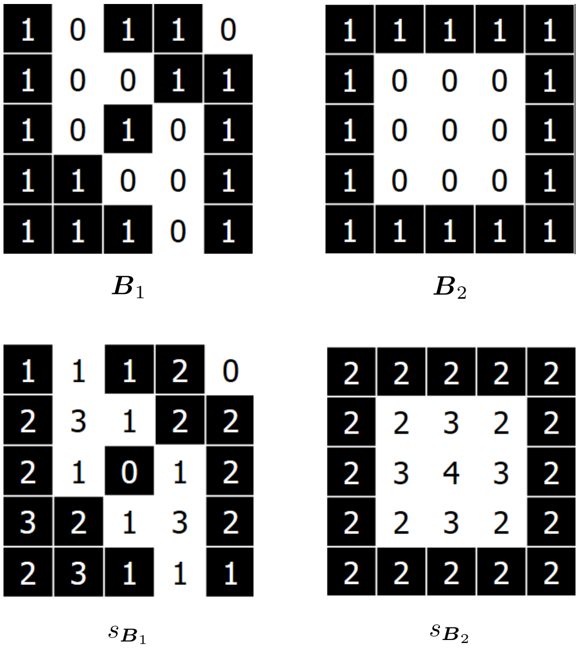
In general, the $\mathcal{RKHS}$ methods tend to yield dense estimation, i.e., the values are non zeros almost everywhere in the image domain. The population-level masking image $\boldsymbol{B}$ is introduced to achieve a desired sparse structure. Specifically, the regions with $\boldsymbol{B}(t) = 1$ capture the effective (non-zero) parts of $\boldsymbol{\beta}$, and the ones with $\boldsymbol{B}(t) = 0$ represent the ineffective regions, i.e., $\boldsymbol{\beta}(t) = 0$ if $\boldsymbol{B}(t) = 0$. We assume that this masking image $\boldsymbol{B}$ follows a Potts model (Besag, 1986; Zhang et al., 2001), which measures the probability of binary patterns in the image domain $\mathcal{D}$. Its probability mess function (PMF) is given as follows,

$$p(\boldsymbol{B}|\tau) = \exp\left\{\tau \sum_{t\in\mathcal{D}} s_{\boldsymbol{B}}\left(t\right)\right\} C(\tau). \tag{3.7}$$

Here $s_{\boldsymbol{B}}(t)$ is the local similarity score of the binary image $\boldsymbol{B}$ at location $t \in \mathcal{D}$, which is given by

$$s_{\boldsymbol{B}}\left(t\right) = \sum_{t'\in\mathcal{N}_t} \delta\left(\boldsymbol{B}(t'), \boldsymbol{B}(t)\right), \tag{3.8}$$

where $\delta(\cdot,\cdot)$ is the Kronecker function, i.e., $\delta(s,t) = 1$ if $s = t$ and 0 if $s \neq t$, and $\mathcal{N}_t$ defines all the one-step neighborhoods of $t$. In particular, we use the 4 adjacent pixels and the 6 adjacent voxels as the one-step neighborhoods for the 2D and 3D images respectively. The parameter $\tau$ controls the level of spatial smoothness and local similarity of $\boldsymbol{B}$ and the normalizing factor $C(\tau)$ is introduced to ensure $P(\boldsymbol{B}|\tau)$ defines a proper PMF, i.e., $C(\tau) = 1/\sum_{\boldsymbol{B}\in\mathcal{D}_{\boldsymbol{B}}} P(\boldsymbol{B}|\tau)$, where $\mathcal{D}_{\boldsymbol{B}}$ denotes the set of all binary images over the image domain $\mathcal{D}$. The example in Figure 3.2 illustrates two binary images $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ with exactly same number of 0 and 1's, but different



| | | | | |
|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 |

$\boldsymbol{B}_1$

| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

$\boldsymbol{B}_2$

| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 2 | 0 |
| 2 | 3 | 1 | 2 | 2 |
| 2 | 1 | 0 | 1 | 2 |
| 3 | 2 | 1 | 3 | 2 |
| 2 | 3 | 1 | 1 | 1 |

$s_{\boldsymbol{B}_1}$

| | | | | |
|---|---|---|---|---|
| 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 3 | 2 | 2 |
| 2 | 3 | 4 | 3 | 2 |
| 2 | 2 | 3 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 |

$s_{\boldsymbol{B}_2}$

**Figure 3.2:** Example of binary images and their associated local similarity scores. The top row displays the binary images $\boldsymbol{B}_1(t)$ and $\boldsymbol{B}_2(t)$, which have exactly same numbers of 0's and 1's. The bottom row displays their associated local similarity scores $s_{\boldsymbol{B}_1}(t)$ and $s_{\boldsymbol{B}_2}(t)$.

patterns. Their local similarity scores $s_{\boldsymbol{B}_1}$ and $s_{\boldsymbol{B}_2}$ are computed according to Equation (3.8). The associated Potts likelihood are computed using Equation (3.7). It can be calculated that $P(\boldsymbol{B}_2) \propto \exp(\tau \times 56) > P(\boldsymbol{B}_1) \propto \exp(\tau \times 44)$ and the PMF of $\boldsymbol{B}_2$ has larger value than $\boldsymbol{B}_1$ as a result.

Under the Potts model assumption, with large probability the binary image $\boldsymbol{B}$ will partition the whole image domain $\mathcal{D}$ into several disjoint non-zeros regions, denoted as $\mathcal{R}_k \subset \mathcal{D}$ for $k = 1, \ldots, K$, i.e.,

$$\boldsymbol{B}(t) = \begin{cases} 1 & \text{if } t \in \cup_k \mathcal{R}_k, \\ 0 & \text{if } t \in \mathcal{D} / \cup_k \mathcal{R}_k. \end{cases} \tag{3.9}$$

We will use these region $\mathcal{R}_k$'s to characterize the potential disease-related regions or the possible lesion locations in the medical images. Since the population disease map is construct within the support regions of $\boldsymbol{B}$, the overall coefficient image $\boldsymbol{\beta}$ will yield spatial smoothness within $\mathcal{R}_k$'s and spatial sparsity outsides.

#### 3.2.2.2 Individual disease maps

In the SVSIR model 3.2, the heterogeneity is characterized by the subject-specific coefficient $\boldsymbol{\beta}_i$'s, and each $\boldsymbol{\beta}_i$ is constructed by applying an masking image $\boldsymbol{B}_i$ on the homogeneous coefficient $\boldsymbol{\beta}$, i.e., $\boldsymbol{\beta}_i = \boldsymbol{B}_i \circ \boldsymbol{\beta}$. These heterogeneous coefficient images capture the intrinsic biological structures through the spatially smoothing homogeneous map $\boldsymbol{\beta}$ and incorporate heterogeneity structure via the masking images $\boldsymbol{B}_i$'s.

The heterogeneous masking images are constructed by activating different detected regions, defined in Equation (3.9) in the homogeneous mask $\boldsymbol{B}$. More specifically,

$$\boldsymbol{B}_i = \sum_{k=1}^{K} I_{ik} \boldsymbol{R}_k, \tag{3.10}$$

where $I_{ik} \in \{0,1\}$ indicates whether region $\mathcal{R}_k$ is active or not for the $i$-th subject and $\boldsymbol{R}_k$ denotes the support function for region $\mathcal{R}_k$, i.e,

$$\boldsymbol{R}_k(t) = \begin{cases} 1 & \text{if } t \in \mathcal{R}_k, \\ 0 & \text{if } t \in \mathcal{D}/\mathcal{R}_k. \end{cases} \tag{3.11}$$

In summary, our model contains two sets of coefficients: the population coefficients of $\boldsymbol{\beta}$ and $\boldsymbol{B}$, and individual-level coefficients $\boldsymbol{B}_i$'s. The homogeneous disease map $\boldsymbol{\beta}$ characterizes the overall regression relationship between the disease status $y_i$'s and the medical images $\boldsymbol{X}_i$'s. The masking image $\boldsymbol{B}$ reinforces the spatial smoothness and sparseness of $\boldsymbol{\beta}$ and captures the possible disease-related regions $\mathcal{R}_k$'s. The individual-level binary image $\boldsymbol{B}_i$'s determine the active regions for each subject, and captures the heterogeneity structure of the population.

### 3.3 Estimation and prediction

Our next problem of interest is to estimate the unknown coefficients in the model and establish the prediction rules. The population disease map $\boldsymbol{\beta}$ is estimated in a functional regression model. The distribution of the homogeneous masking image $\boldsymbol{B}$ can be obtained using a maximum of a posterior probability. The individual-level parameters $\boldsymbol{B}_i = \sum_{k=1}^{K} I_{ik} \boldsymbol{R}_k$, are estimated by determining region assignment $I_{ik}$. The estimation of homogeneous and heterogeneous coefficients are proceeded iteratively. The prediction rule is constructed via a pattern matching process. We will discuss the details in this section.

### 3.3.1 Homogeneous disease map

In the SVSIR model, the heterogeneity is captured by the subject-specific masking image $\boldsymbol{B}_i$'s as stated in Equation (3.4). Thus, give the binary image $\boldsymbol{B}_i$'s, we can proceed a feature screening procedure and extract the informative regions in each covariate image. The masked covariate images are defined as $\tilde{\boldsymbol{X}}_i = \boldsymbol{B}_i \circ \boldsymbol{X}_i$. The whole regression model is reduced to the following form,

$$y_i = \beta_0 + \int_{t \in \mathcal{D}} \tilde{\boldsymbol{X}}_i(t) \boldsymbol{\beta}(t) dt + \epsilon_i \quad \text{for each } i \in \{1, \ldots, n\}. \tag{3.12}$$

Note that model (3.12) is a homogeneous functional regression defined in Equation (3.1), with covariates $\tilde{\boldsymbol{X}}_i$'s and response $y_i$'s. To simplify the notation, we center all the covariates and responses so that the intercept term $\beta_0$ can be dropped in the derivation. Since coefficient image $\boldsymbol{\beta}$ resides in a $\mathcal{RKHS}$ induced by the radial based kernel $K$ given by Equation (3.6), we can estimate the coefficient image using a kernel ridge regression method in (Yuan et al., 2010). In particular, the parameters can be estimated by solving the following empirical risk minimization (ERM),

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{f} \in \mathcal{H}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \left( y_i - \int_{t \in \mathcal{D}} \tilde{\boldsymbol{X}}_i(t) \boldsymbol{f}(t) dt \right)^2 + \lambda J(\boldsymbol{f}), \tag{3.13}$$

where the first term is the squared loss which ensures the goodness of fit on the training data, $J(\boldsymbol{\beta})$ represents the smoothness penalty and $\lambda$ is the parameter that controls the level of the smoothness and the complexity of the estimated coefficient.

According to the representer theorem (Wahba, 1990), there exists an $c = (c_1, \ldots, c_n)' \in \mathbb{R}^n$ such that the solution to (3.13) can be expressed as follows,

$$\hat{\boldsymbol{\beta}}(t) = \sum_{i=1}^{n} c_i \int_{s \in \mathcal{D}} K(t, s) \tilde{\boldsymbol{X}}_i(s) ds. \tag{3.14}$$

The roughness penalty can be expressed as $J(\boldsymbol{\beta}) = c'\Sigma c$, where $\Sigma$ is a $n$ by $n$ Gram matrix with

$$\Sigma_{ij} = \iint \tilde{\boldsymbol{X}}_i(s) K(s,t) \tilde{\boldsymbol{X}}_j(t) dsdt \quad \text{for } i,j \in \{1,\ldots,n\}.$$

Therefore, the ERM in (3.13) is equivalent to the following problem:

$$\begin{aligned}
\hat{c} &= \underset{c\in\mathbb{R}^n}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \int_{t\in\mathcal{D}} \tilde{\boldsymbol{X}}_i(t) \sum_{j=1}^{n} c_j \int_{s\in\mathcal{D}} K(t,s)\tilde{\boldsymbol{X}}_j(s) dsdt\right)^2 + \lambda c'\Sigma c. \\
&= \underset{c\in\mathbb{R}^n}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{n} c_j \int\int \tilde{\boldsymbol{X}}_i(t) K(t,s)\tilde{\boldsymbol{X}}_j(s) dsdt\right)^2 + \lambda c'\Sigma c. \\
&= \underset{c\in\mathbb{R}^n}{\operatorname{argmin}} \|Y - \Sigma c\|^2 + n\lambda c'\Sigma c,
\end{aligned}$$

where $Y = [y_1, \ldots, y_n]'$ denotes the response vector.

This is a Tikhonov regularization (Tikhonov, 1943) which has a close form solution:

$$\hat{c} = \left(\Sigma^2 + n\lambda\Sigma\right)^{-1}\Sigma Y.$$

Finally, we recover the coefficient image by plugging the estimation of $c$ into Equation (3.14), i.e.

$$\hat{\boldsymbol{\beta}}(t) = \sum_{i=1}^{n} \hat{c}_i \int_{s\in\mathcal{D}} K(t,s)\tilde{\boldsymbol{X}}_i(s) ds \tag{3.15}$$

### 3.3.2 Homogeneous region detection

The population-level masking image $\boldsymbol{B}$ follows a Potts model with the probability mass function in Equation (3.7). We apply the method of maximizing the posterior probability (Bassett and Deride, 2016), and estimate $\hat{\boldsymbol{B}}$ by solving the following optimization of the joint likelihood of $Y$

and $\boldsymbol{B}$:

$$\begin{aligned}\hat{\boldsymbol{B}} &= \operatorname*{argmax}_{\boldsymbol{B}\in\mathcal{D}_{\boldsymbol{B}}} L(Y, \boldsymbol{B}; \hat{\boldsymbol{\beta}}, \tilde{\boldsymbol{X}}_i\text{'s}, \tau, \sigma^2) \\ &= \operatorname*{argmax}_{\boldsymbol{B}\in\mathcal{D}_{\boldsymbol{B}}} \prod_{i=1}^{n} \left( \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{r_i^2}{2\sigma^2} \right\} \right) \cdot \exp\left\{ \tau \sum_{t\in\mathcal{D}} s_{\boldsymbol{B}}\left(t\right) \right\} C(\tau) \\ &= \operatorname*{argmin}_{\boldsymbol{B}\in\mathcal{D}_{\boldsymbol{B}}} \sum_{i=1}^{n} r_i^2 - \tau \sum_{t\in\mathcal{D}} s_{\boldsymbol{B}}\left(t\right) + C, \end{aligned} \tag{3.16}$$

where $\mathcal{D}_{\boldsymbol{B}} = \{0,1\}^{\otimes|\mathcal{D}|}$ represents the set of all possible binary images on $\mathcal{D}$, and $r_i = y_i - \int_{t\in\mathcal{D}} \tilde{\boldsymbol{X}}_i(t)\boldsymbol{B}(t)\hat{\boldsymbol{\beta}}(t)$ corresponds to the residual terms and $C$ is a normalizing factor that does not depend on $\boldsymbol{B}$.

This is a non-convex integer programming and the feasible set $\mathcal{D}_{\boldsymbol{B}} = \{0,1\}^{\otimes|\mathcal{D}|}$ is countably finite. One approach is to enumerate all the possible element in $\mathcal{D}_{\boldsymbol{B}}$ at the cost of $\mathcal{O}(2^{|\mathcal{D}|})$ operations. This is in general unachievable when the size of the image is large. Instead, we adopt an Iterative Conditional Modes (ICM) algorithm (Besag, 1986), which uses a greedy iterative strategy to search for a local minimal. Convergence is usually achieved after a few iterations with a complexity of $\mathcal{O}(|\mathcal{D}|)$ operations.

Due to the Potts prior, $\hat{\boldsymbol{B}}$ usually yields a pattern with disjoint regions. We label those regions as $\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_K$, and decompose $\hat{\boldsymbol{B}}$ into $K$ binary images $\hat{\boldsymbol{R}}_k$'s as defined in Equation (3.11).

The parameter $\tau$ controls the level of spatial smoothness and local similarity. Its maximum likelihood estimation is generally difficult to compute due to the normalization factor $C(\tau)$. Thus, we estimate it by maximizing a pseudo-likelihood given as follows,

$$\begin{aligned} L(\hat{\boldsymbol{B}};\tau) &= \prod_{t\in\mathcal{D}} PL(\hat{\boldsymbol{B}}(t)|\hat{\boldsymbol{B}}) \\ &= \prod_{t\in\mathcal{D}} \frac{\exp\Big(\tau \sum_{s\in\mathcal{N}(t)} \delta(\hat{\boldsymbol{B}}(s), \hat{\boldsymbol{B}}(t)\Big)}{\exp\Big(\tau \sum_{s\in\mathcal{N}(t)} \delta(\hat{\boldsymbol{B}}(s), 1\Big) + \exp\Big(\tau \sum_{s\in\mathcal{N}(t)} \delta(\hat{\boldsymbol{B}}(s), 0\Big)}. \end{aligned}$$

The parameter $\hat{\tau}$ is estimated by setting the derivative to zero, i.e.,

$$\frac{\partial \ln\Big(L(\hat{\boldsymbol{B}};\tau)\Big)}{\partial\tau} = 0$$

### 3.3.3 Heterogeneous regions assignments

In this section, we will introduce the heterogeneous coefficient estimation. As defined in Equation (3.4), our model captures the heterogeneity through the individual-level binary masking image $\boldsymbol{B}_i = \sum_{k=1}^{K} I_{ik}\boldsymbol{R}_k$. Give the estimation of the homogeneous masking image $\hat{\boldsymbol{B}}$, it suffices to determine the indicators $I_{ik}$'s. We assume that the region assignment for each subjects are independent, thus they can be estimated component-wisely by minimizing the prediction squared error, i.e.,

$$\begin{aligned}[\hat{I}_{i1}, \ldots, \hat{I}_{iK}] &= \underset{\{0,1\}^{\otimes K}}{\operatorname{argmin}} \left( y_i - \sum_{k=1}^{K} I_{ik} \int \boldsymbol{X}_i(t)\hat{\boldsymbol{R}}_k(t)\hat{\boldsymbol{\beta}}(t)dt \right)^2 \qquad (3.17)\\ &= \underset{\{0,1\}^{\otimes K}}{\operatorname{argmin}} |y_i - \sum_{k=1}^{K} I_{ik}\hat{\mu}_{ik}|,\end{aligned}$$

where $\hat{\mu}_{ik} = \int \boldsymbol{X}_i(t)\hat{\boldsymbol{R}}_k(t)\hat{\boldsymbol{\beta}}(t)dt$.

Problem (3.17) is a binary integer programming problem. Due to the effect of the Potts prior, the total number of regions ($K$) detected by $\hat{\boldsymbol{B}}$ cannot be too large. Thus we can do a enumerative search on the feasible set of $\{0,1\}^{\otimes K}$ to find the best solution. The optimization for each subject can be conducted efficiently in parallel to save the computational time.

In practice, this algorithm tends over-fit by assigning false positive regions with weak signal, i.e. $|\hat{\mu}_{ik}|$ is small. In order to avoid this issue, we impose a penalty on the region assignments $I_{ik}$'s, while leads to the following optimization problem,

$$[\hat{I}_{i1}, \ldots, \hat{I}_{iK}] = \underset{\{0,1\}^{\otimes K}}{\operatorname{argmin}} |y_i - \sum_{k=1}^{K} I_{ik}\mu_{ik}| + \lambda_s \sum_{k=1}^{K} |I_{ik}|, \qquad (3.18)$$

where $\lambda_s$ is a parameter controls the sparsity level of the region assignment. In particular, by imposing such penalty, we need to achieve at least a decrement of $\lambda_s$ in terms of the loss function value, in order to assign the associated region as active. In our implementation, we determine its value according to the variation of training samples, i.e., $\lambda_s = 0.05 * \mathrm{Var}(Y)$.

In summary, the whole estimation procedure can be illustrated by the flowchart in Figure 3.3. The orange arrow characterize the estimation procedure for population-level coefficient $\boldsymbol{\beta}$ and $\boldsymbol{B}$; the green arrow denotes the individual-level estimation. In particular, based on the masked

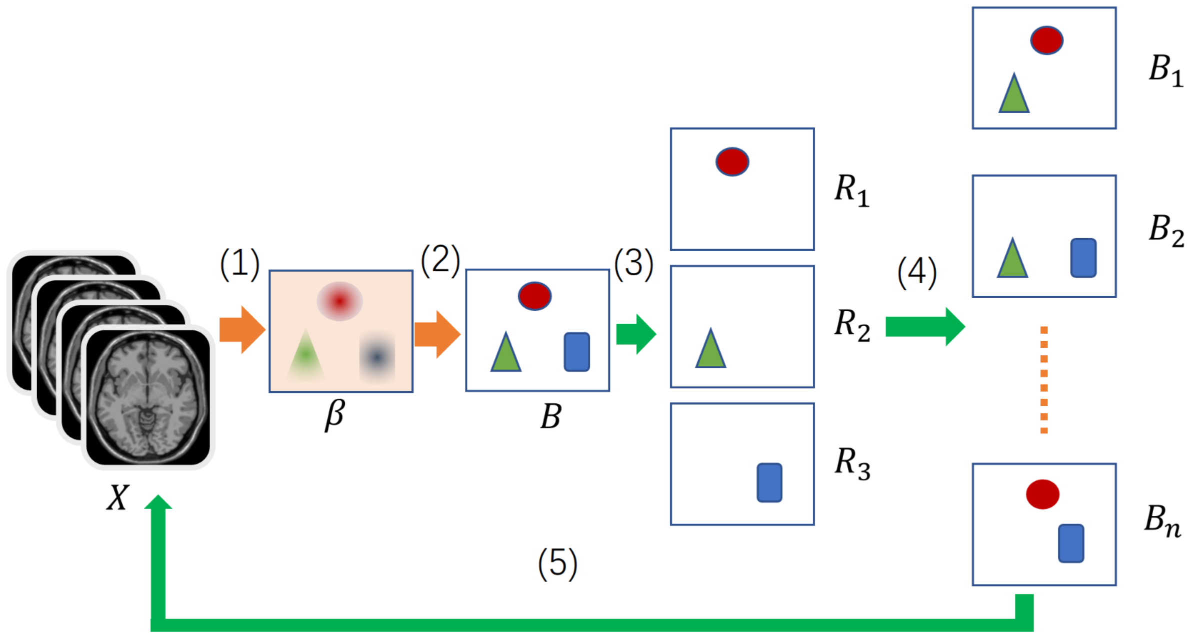


**Figure 3.3:** The flow chart of the estimation procedure.

covariate image $\tilde{\boldsymbol{X}}_i$'s, we estimate $\boldsymbol{\beta}$ in step (1) according to Equation (3.15); and then in step (2) we update $\boldsymbol{B}$ by maximize the joint likelihood defined in Equation (3.16). These two steps only estimate the population-level coefficients and do not change the individual-level region assignments. In step (3), $\boldsymbol{R}_k$ is extracted based on the estimation of $\boldsymbol{B}$ and in step (4), the regions assignments $I_{ik}$'s are obtained by minimizing the training loss according to Equation (3.17). Based on the region assignments, we update the heterogeneous hidden layer $\boldsymbol{B}_i$'s, and reconstruct the covariates $\tilde{\boldsymbol{X}}_i$'s in step (5). The whole estimation procedure is conducted sequentially and iteratively until a convergence criteria is met. In this project, we terminate the algorithm when the change of individual-level coefficient $\boldsymbol{\beta}_i$'s are less then a preset threshold .

### 3.3.4 Tuning parameter selection

The population-level of our model includes a kernel ridge regression estimation on the $\mathcal{RKHS}$. There are two tuning parameters involved: the band width $\sigma$ for the radial based kernel in Equation (3.6), and the ridge penalty level $\lambda$ in Equation (3.13). We use the Bayesian Information Criteria (BIC) as the guidance to select the parameters.

According to the Stein's unbiased risk estimation (SURE) theory (Stein, 1981), the degrees of freedom is defined as:

$$df = \sum_{i=1}^{n} cov(\hat{\mu}_i, y_i) = \sum_{i=1}^{n} \partial\hat{\mu}_i/\partial y_i.$$

In our model, the estimation step for $\boldsymbol{\beta}$ can be expressed as follows:

$$\hat{Y} = \Sigma(\Sigma^2 + n\lambda\Sigma)^{-1}\Sigma Y = HY,$$

where $\Sigma_{i,j} = \iint \tilde{\boldsymbol{X}}_i(s)K(s,t)\tilde{\boldsymbol{X}}_j(t)dsdt$ and $\tilde{\boldsymbol{X}}_i = \boldsymbol{X}_i \circ \boldsymbol{B}_i$. Thus, the degrees of freedom can be approximated by $df = trace(H)$, and the BIC is therefore given by

$$BIC = \log(n) \times df + n\log\left(\frac{RSS}{n}\right),$$

where $RSS$ represents the residual sum of square which is given by $RSS = \sum_{i=1}^{n}(\hat{y}_i - y)^2$. The parameter $\sigma$ and $\lambda$ is chosen to minimize the $BIC$.

Since the estimation procedure is proceeded sequentially in each iteration and the estimation of $\hat{\boldsymbol{B}}$ and $\hat{\boldsymbol{B}}_i$'s do not involve the tuning process, instead of selecting the parameters after the whole estimation process, we embed the tuning procedure in the kernel ridge regression only, i.e., step (2) in Figure 3.3. Thus, the whole estimation procedure can be viewed as a tuning free algorithm.

### 3.3.5 Prediction

The prediction for the models involving hidden variables or unobserved data is still an open question. The mixture regression models mainly use the posterior probability $\hat{\pi}_k$ to do the weighted pooling, i.e., $y^* = \sum_{k=1}^{K}\hat{\pi}_k\boldsymbol{X}^*\boldsymbol{\beta}_k$. This in general yields underperformed prediction accuracy at the individual-level, since it doesn't take the information of $X^*$ into consideration and treat each subject as identically distributed. (Hoshikawa, 2013) proposed a jointly mixture regression model, which incorporate the joint distribution between the covariates $\boldsymbol{X}$ and the response $y$ in the prediction. Particularly, they use a different pooling strategy, i.e., $y^* = \sum_{k=1}^{K}\hat{\pi}_k(\boldsymbol{X}^*)\boldsymbol{X}^*\boldsymbol{\beta}_k$, where $\hat{\pi}_k(\boldsymbol{X}^*)$

denotes the estimated probability of $\boldsymbol{X}^*$ belonging to group $k$. The improvement of the prediction accuracy relies the Kullback-Leibler divergence of the component-wise distribution.

Instead of being modeled as probabilistic distributions, the heterogeneity in the propose SVSIR model is characterized by the subject-specific masking image. For a new subject, we first determine its regions assignment, and then construct the individual-level coefficient $\boldsymbol{B}^*$. The predicted response is then given by

$$y^* = \int_{t \in \mathcal{D}} \boldsymbol{X}^*(t)\boldsymbol{B}^*(t)\hat{\boldsymbol{\beta}}(t)dt.$$

In the image analysis, the covariate images usually contains different signals in the regions that are related to the response from irrelevant regions. The regional signals may not strong enough to be directly detected in the raw image, but may be identified through the regression models. In the proposed SVSIR model, both the hidden binary image $\boldsymbol{B}$ and the disease map $\boldsymbol{\beta}$ can screen out the irrelevant regions thus enhance the predictive signals. Based on the region assignments $I_{ik}$ in the training data, we can detect the two patterns in each detected region $\mathcal{R}_k$: active signals (e.g., lesions tissue) from the subjects with $I_{ik} = 1$; and inactive signals (e.g., normal tissues) from the subjects with $I_{ik} = 0$. In particular, the two patterns can be computed as follows,

$$\begin{aligned} &\text{Active:} && \boldsymbol{P}_{k1} = \frac{\sum_{i=1}^n I_{ik}\boldsymbol{X}_i \circ \boldsymbol{R}_k}{\sum_{i=1}^n I_{ik}} \\ &\text{Inactive:} && \boldsymbol{P}_{k0} = \frac{\sum_{i=1}^n (1 - I_{ik})\boldsymbol{X}_i \circ \boldsymbol{R}_k}{\sum_{i=1}^n (1 - I_{ik})}. \end{aligned}$$

Based on the regional patterns in the training data, we can build a classifier for each region, and use the classifiers to determine whether to assign active or inactive regions for a new subject. In particular, we use the distance between regional signal $\boldsymbol{X}^* \circ \boldsymbol{R}_k$ and the active and inactive patterns as the classification rule to determine the region assignment, i.e.

$$I_k^* = \mathbf{1}\left\{d(\boldsymbol{X}^* \circ \boldsymbol{R}_k, \boldsymbol{P}_{k1}) < d(\boldsymbol{X}^* \circ \boldsymbol{R}_k, \boldsymbol{P}_{k0})\right\}, \tag{3.19}$$

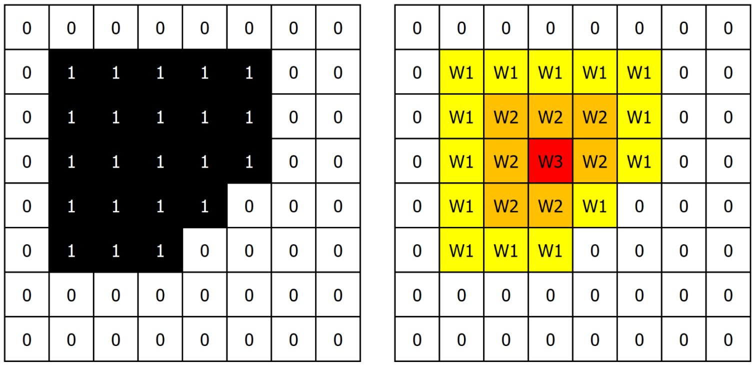
where $d(\cdot)$ measures the weighted distance between two image signals. With this region assignment, the prediction rule is summarized as follows,

$$y^* = \sum_{k=1}^{K} I_k^* \int_{t \in \mathcal{R}_k} \boldsymbol{X}^*(t)\hat{\boldsymbol{\beta}}(t)dt.$$

Here we use a shape based weighted average of the squared pixel/voxel-wise difference to compute the distance, i.e.,

$$d(\boldsymbol{X} \circ \boldsymbol{R}_k, \boldsymbol{P}_{kl}) = \sqrt{\int_{t \in \mathcal{R}_k} (\boldsymbol{X}(t) - \boldsymbol{P}_{kl}(t))^2 W(t)dt} \quad \text{for } l = 1, 0.$$

The signals near the center of the region have more weights, and those near boundary have less weights. This weight is introduced mainly for the consideration of the errors in the region estimations. In general, the estimated regions can cover the center with a high probability, but may yield more errors in detecting the boundaries. This strategy can effectively reduce the biasness from the estimation error. An graphical illustration of the shape-based weights $W(t)$ is given in Figure 3.4.



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | W1 | W1 | W1 | W1 | W1 | 0 | 0 |
| 0 | W1 | W2 | W2 | W2 | W1 | 0 | 0 |
| 0 | W1 | W2 | W3 | W2 | W1 | 0 | 0 |
| 0 | W1 | W2 | W2 | W1 | 0 | 0 | 0 |
| 0 | W1 | W1 | W1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 3.4:** An example of the shape-based weights. The left panel is a binary image, and the right panel is the associated weight matrix, with $W1 < W2 < W3$.

## 3.4 Theoretical properties

In this section, we investigate the theoretical properties of the proposed SVSIR. The model contains several estimation procedures, which increase the difficulty to study the theoretical properties of the method. It is convenient to simplify the method and check the performance in certain scenarios.

We assume the hidden binary region map $\boldsymbol{B}$ in Equation (3.5) is known, and the individual-level region assignments $\mathrm{I}_{ik}$ in Equation (3.10) are also given, and we exam the theoretical properties of the estimation of $\hat{\boldsymbol{\beta}}$ in Equation (3.13). Based on the training sample and the given condition, we have $n$ independent samples, $\{\left(\tilde{\boldsymbol{X}}_i, Y_i\right) : i = 1, \ldots, n\}$. The estimation accuracy can be measured by the excess risk:

$$\begin{aligned}\mathcal{E}(\hat{\boldsymbol{\beta}}) &= \mathbb{E}^* \left[Y^* - \int \tilde{\boldsymbol{X}}^*(t)\hat{\boldsymbol{\beta}}(t)dt\right]^2 - \mathbb{E}^* \left[Y^* - \int \tilde{\boldsymbol{X}}^*(t)\boldsymbol{\beta}(t)dt\right]^2 \\ &= E^* \left[\int \tilde{\boldsymbol{X}}^*(t) \left(\hat{\boldsymbol{\beta}}(t) - \boldsymbol{\beta}(t)\right) dt\right]^2.\end{aligned}$$

We will investigate some asymptotic properties of excess risk in this section.

Let $\mathcal{L}^2$ denotes the space of square integrable functions. For $f, g \in \mathcal{L}^2$, their inner product is given by,

$$\langle f, g\rangle_{\mathcal{L}^2} = \iint f(s)g(t)dsdt.$$

For a real valued kernel function $K$: $\mathcal{D} \times \mathcal{D} \to \mathbb{R}$, denote the $L_K$ as the integral operator, i.e.,

$$L_K(f)(\cdot) = \langle K(s, \cdot), f\rangle_{\mathcal{L}^2} = \int K(s, \cdot)f(s)ds.$$

For $f, g \in \mathcal{L}^2$, let $\langle f, g\rangle_K$ denote the inner product induced by the kernel function $K$, i.e.,

$$\langle f, g\rangle_K = \iint f(s)K(s,t)g(t)dsdt.$$

The norm associated with such inner product are defined as $\|f\|_K^2 = \langle f, f\rangle_K$. Without specification in the subscript, $\|\cdot\|$ denote the $\mathcal{L}^2$-norm.

For the random process $\tilde{\boldsymbol{X}}$, we denote its covariance operator as follows,

$$C(s,t) = \mathbb{E}\left[(\tilde{\boldsymbol{X}}(s) - \mathbb{E}\tilde{\boldsymbol{X}}(s))(\tilde{\boldsymbol{X}}(t) - \mathbb{E}\tilde{\boldsymbol{X}}(t))\right],$$

It is obvious that

$$\begin{aligned}\mathcal{E}(\hat{\boldsymbol{\beta}}) &= \iint \left(\hat{\boldsymbol{\beta}}(s) - \boldsymbol{\beta}(s)\right) C(s,t) \left(\hat{\boldsymbol{\beta}}(t) - \boldsymbol{\beta}(t)\right) dsdt \\ &= \|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_C^2.\end{aligned}$$

According to the spectral theorem, we can construct an eigen-decomposition for $C$ with a set of orthonormal eigenfunctions $\left\{\psi_k^C : k \geq 1\right\}$ and a sequence of non-increasing eigenvalues $\left\{\lambda_1^C \geq \lambda_2^C \geq \dots\right\}$, i.e.,

$$C(s,t) = \sum_{k=1}^{\infty} \lambda_k^C \psi_k^C(s) \psi_k^C(t).$$

The eigen-decomposition of the kernel $K$ can be defined similarly.

For two sequences of positive real numbers, $\{a_k\}$ and $\{b_k\}$, we denote $a_k \asymp b_k$ if $\frac{a_k}{b_k}$ is bounded from above and from zero for all $k > 1$, i.e., $0 < L \leq \frac{a_k}{b_k} \leq U < \infty$.

In this section, we will investigate the asymptotic properties of the estimator $\hat{\boldsymbol{\beta}}$ in terms of $\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_C^2$. We need the following assumptions to develop our theoretical results:

(A.1) The sample surface/volume of the covariate $\tilde{\boldsymbol{X}}$ follows a Gaussian random field over the image domain $\mathcal{D}$.

(A.2) The eigenvalues for the covariance operator $C$ of the covariate $\tilde{\boldsymbol{X}}$ satisfy $\lambda_k^C \asymp k^{-2a}$ with $a > 1/2$.

(A.3) $L_C(\boldsymbol{\beta}) \neq 0$ for all $\boldsymbol{\beta} \in \mathcal{H}$ and $\boldsymbol{\beta} \neq 0$.

(A.4) The eigenvalues for the kernel $K$ of the space $\mathcal{H}$ satisfy $\lambda_k^K \asymp k^{-2b}$ with $b > 1/2$.

(A.5) The functions $K$ and $C$ can be simultaneously decomposed with eigenvalues $\lambda_k^* \asymp \lambda_k^K \lambda_k^C$ and eigenfunctions $\psi_k^*$, where $\nu_k = \left(1 + \frac{\lambda}{\lambda_k^*}\right)^{-1}$ the eigenvalues of $R^{1/2} C R^{1/2}$ associated with the eigenfunctions $\xi_k$, and $R$ represents the reproducing kernel associated with the following norm:

$$\|f\|_R^2 = \|f\|_C^2 + \lambda J(f).$$

Note that the assumption (A.1) ensures the boundedness of moments of the covariate; assumption (A.2) specifies the smoothness of the sample surface/volume; conditions (A.3)-(A.5) are required for the simultaneously eigen-decomposition of functions $K$ and $C$.

**Theorem 1.** *Under assumptions (A.1)-(A.5), with the roughness penalty parameter $\lambda$ satisfying $\lambda \asymp n^{-\frac{2(a+b)}{2(a+b)+1}}$,*

$$\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_C^2 = O_p\left(n^{-\frac{2(a+b)}{2(a+b)+1}}\right).$$

Note that Theorem 1 indicates that the convergence rate of the estimator $\hat{\boldsymbol{\beta}}$ (in terms of the excess risk) is determined by the smoothness of the covariate images, the decay rate of the kernel of the Hilbert space and the alignment of the eigen-system of the covariance function $C$ and the kernel $K$. More specifically, if the eigen-system of the two functions can be perfectly aligned, i.e., can be decomposed with the same set eigenfunctions, we will have the following result,

**Corollary 1.** *If $\psi_k^K = \psi_k^C$ for all $k \geq 1$, under the assumptions of Theorem 1, we have*

$$\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|^2 = O_p\left(n^{-\frac{2b}{2(a+b)+1}}\right),$$

*with $\lambda \asymp n^{-\frac{2(a+b)}{2(a+b)+1}}$.*

This is a special case when the $\mathcal{RKHS}$ method coincides with to the functional PCA based regression (Cai et al., 2006; Hall et al., 2007). It indicates that the convergence rate of the estimation error increases as the decay of $\lambda_k^C$'s getting faster.

### 3.5 Simulation studies

In this section, we illustrate the numerical performance of SVSIR and conduct multiple simulation studies in varies settings. The covariate images $\boldsymbol{X}_i$'s are generated from a 2D Gaussian random field, with their boundaries constrained as zeros. The population-level coefficient image $\boldsymbol{\beta}$ is generated according to the Gaussian functions centered at different locations in the image domain. The number of non-zero regions are 2, 3 and 5 in the Scenarios 1, 2 and 3 respectively. The plots of these coefficient images are given in Figure 3.5.

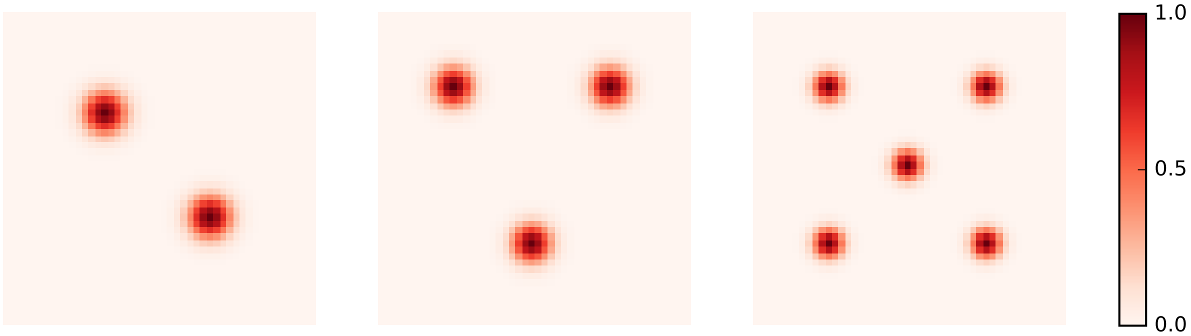


**Figure 3.5:** The plots for the homogeneous coefficient images $\boldsymbol{\beta}$ in Scenarios 1, 2 and 3 respectively.

In each scenario, every individual-level coefficient $\boldsymbol{\beta}_i$ has at least one active region chosen randomly from the non-zeros regions in the homogeneous coefficient images. The other regions are assigned to be active with probability 0.2. In each active region $\mathcal{R}_k$, we add some regional signal to the covariate images, denoted as $S_k$. The signal is generated from the corresponding regional signal in the homogeneous coefficient image. The strength of the signals is controlled by the regional signal-to-noise ratio (denoted as regional SNR or RSNR), i.e., $S_k = RSNR * Unif(1, 1.5) * \boldsymbol{\beta} * \boldsymbol{R}_k$. These extra signals induce the heterogeneity among the covariates, and the regional SNR controls the level of the heterogeneity, i.e., larger regional SNR indicates more distinguishable patterns between the active and inactive regions. In our simulation studies, we conduct the experiments in each scenario with regional SNR's equal to 0, 0.25, 0.5, 0.75 and 1. These settings cover a wide range of heterogeneity among covariates, from the case of homogeneity to weak, mediate and strong heterogeneity. Figure 3.6 gives an example of the coefficient image and an associated covariate image in the moderate heterogeneity case ($RSNR = 0.5$). There are 3 non-zero regions in the homogeneous coefficient image, and the two on the top are active in the covariate image.

We compare our method with some off-the-shelf high dimensional regression models, which includes ridge regression (RR) (Hoerl and Kennard, 1970), Elastic-Net regression (EN) (Zou and Hastie, 2005), Lasso regression (Lasso) (Tibshirani, 1996), Spatial regularized regression (SREG) (Liu et al., 2018) and the functional linear regression (FLR) (Yuan et al., 2010).

We generate 200 training samples and another independent test set of 200 images for each simulation setting. We used a ten-fold cross validation for the model selection of the methods we are comparing to. To evaluate the performance, we use the Relative Estimation Error (REE) of

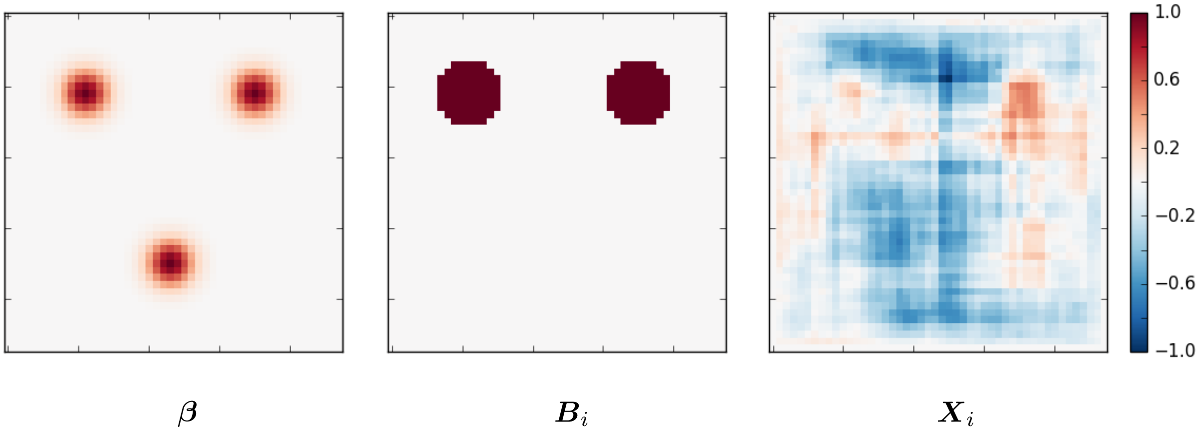


**Figure 3.6:** An example of the synthetic image in the moderate heterogeneity case: the left panel is the coefficient image with 3 non-zero regions. The middle panel is the masking image, indicating that the top two regions are active; and the right panel is the corresponding covariate image $\boldsymbol{X}$ with $RSNR = 0.5$.

the common disease map $\boldsymbol{\beta}$, and the Root Mean Squared Prediction Error (RMSPE) evaluated on the test samples to measure the performance of the models. In particular, their mathematical formulation is given by

$$\begin{aligned} REE &= \frac{\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|^2}{\|\boldsymbol{\beta}\|^2}, \\ RMSPE &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i^* - y_i^*)^2}, \end{aligned} \tag{3.20}$$

where $^*$ indicates the test samples.

The experiment is repeated for 50 times. The mean and standard deviation of the REE and RMSPE are reported in Tables 3.1 and 3.2. Figure 3.7 illustrates the box plot of the REE and RMSPE, with the median and 25th and 75th quantiles displayed.

From the simulation studies, we can see that the proposed SVSIR yields very competitive estimation results in most cases compared to the homogeneous methods. Lasso and Elastic-Net deliver ultra sparse estimation with isolated pixels, while ridge regression yields a overall shrinkage in the estimated coefficient. The SREG and FLR yield clustered patterns in the coefficient image, but contain many false positives in the irrelevant regions. We visualize the estimated coefficient from one replication in Figure 3.8 as an illustration of this phenomenon. Moreover, by neglecting the heterogeneity among the subjects, these methods weaken the signals in the active regions,

**Table 3.1:** The Relative Estimation Error (REE) of the simulation studies. The means from 50 iterations are reported, with standard deviations in parentheses.

| | **Scenario 1** | | | | |
|---|---|---|---|---|---|
| *RSNR* | 0 | 0.25 | 0.5 | 0.75 | 1 |
| RR | 0.95 (0.01) | 0.90 (0.01) | 0.76 (0.02) | 0.61 (0.02) | 0.48 (0.02) |
| EN | 0.97 (0.03) | 0.84 (0.05) | 0.57 (0.06) | 0.46 (0.04) | 0.44 (0.04) |
| Lasso | 0.99 (0.03) | 0.87 (0.07) | 0.60 (0.06) | 0.51 (0.06) | 0.49 (0.05) |
| SREG | 0.35 (0.10) | 0.17 (0.09) | 0.16 (0.06) | 0.18 (0.08) | 0.18 (0.06) |
| FLR | 0.67 (0.04) | 0.49 (0.05) | 0.30 (0.03) | 0.22 (0.02) | 0.18 (0.02) |
| SVSIR | **0.15** (0.09) | **0.14** (0.08) | **0.08** (0.06) | **0.06** (0.04) | **0.06** (0.03) |
| | **Scenario 2** | | | | |
| *RSNR* | 0 | 0.25 | 0.5 | 0.75 | 1 |
| RR | 0.97 (0.01) | 0.89 (0.01) | 0.73 (0.02) | 0.57 (0.03) | 0.44 (0.02) |
| EN | 0.99 (0.02) | 0.87 (0.04) | 0.71 (0.06) | 0.63 (0.05) | 0.60 (0.06) |
| Lasso | 1.00 (0.01) | 0.97 (0.05) | 0.77 (0.07) | 0.72 (0.07) | 0.71 (0.08) |
| SREG | 0.59 (0.12) | 0.24 (0.08) | 0.21 (0.10) | 0.20 (0.06) | 0.20 (0.07) |
| FLR | 0.75 (0.04) | 0.50 (0.04) | 0.30 (0.03) | 0.23 (0.02) | 0.21 (0.02) |
| SVSIR | **0.22** (0.17) | **0.09** (0.03) | **0.06** (0.01) | **0.05** (0.01) | **0.05** (0.01) |
| | **Scenario 3** | | | | |
| *RSNR* | 0 | 0.25 | 0.5 | 0.75 | 1 |
| RR | 0.97 (0.01) | 0.90 (0.02) | 0.73 (0.03) | 0.57 (0.03) | 0.44 (0.03) |
| EN | 1.00 (0.01) | 0.90 (0.04) | 0.84 (0.12) | 0.84 (0.10) | 0.84 (0.05) |
| Lasso | 1.00 (0.01) | 1.02 (0.04) | 1.05 (0.07) | 1.05 (0.07) | 1.06 (0.08) |
| SREG | 0.78 (0.10) | 0.37 (0.12) | 0.25 (0.08) | 0.26 (0.07) | 0.28 (0.07) |
| FLR | 0.80 (0.04) | 0.55 (0.04) | 0.32 (0.03) | 0.26 (0.03) | 0.23 (0.02) |
| SVSIR | **0.70** (0.26) | **0.13** (0.09) | **0.09** (0.03) | **0.11** (0.03) | **0.10** (0.02) |

**Table 3.2:** The Root Mean Square Prediction Error (RMSPE) of simulation studies. The means from 50 iterations are reported, with standard deviations in parentheses.

|  |  |  | **Scenario 1** |  |  |
|---|---|---|---|---|---|
| *RSNR* | 0 | 0.25 | 0.5 | 0.75 | 1 |
| RR | 2.51 (0.14) | 3.34 (0.16) | 4.42 (0.19) | 5.03 (0.28) | 5.08 (0.30) |
| EN | 2.52 (0.14) | 3.20 (0.18) | 3.21 (0.24) | 2.96 (0.20) | 2.83 (0.17) |
| Lasso | 2.53 (0.14) | 3.23 (0.22) | 3.24 (0.24) | 3.01 (0.21) | 2.89 (0.17) |
| SREG | 1.94 (0.14) | 2.04 (0.14) | 2.19 (0.18) | 2.25 (0.16) | 2.28 (0.18) |
| FLR | 2.28 (0.12) | 2.57 (0.15) | 2.63 (0.17) | 2.56 (0.15) | 2.40 (0.16) |
| SVSIR | **2.20** (0.14) | **2.08** (0.17) | **1.50** (0.18) | **1.19** (0.09) | **1.16** (0.08) |
|  |  |  | **Scenario 2** |  |  |
| *RSNR* | 0 | 0.25 | 0.5 | 0.75 | 1 |
| RR | 2.71 (0.14) | 4.08 (0.18) | 5.59 (0.34) | 6.20 (0.38) | 6.24 (0.42) |
| EN | 2.71 (0.15) | 4.03 (0.19) | 4.41 (0.29) | 4.15 (0.31) | 3.99 (0.27) |
| Lasso | 2.71 (0.15) | 4.18 (0.25) | 4.46 (0.30) | 4.27 (0.32) | 4.13 (0.28) |
| SREG | **2.38** (0.16) | 2.66 (0.20) | 2.94 (0.22) | 3.02 (0.22) | 3.07 (0.24) |
| FLR | 2.59 (0.13) | 3.14 (0.16) | 3.32 (0.18) | 3.22 (0.18) | 3.22 (0.21) |
| SVSIR | 2.74 (0.19) | **2.67** (0.21) | **1.91** (0.30) | **1.36** (0.09) | **1.29** (0.10) |
|  |  |  | **Scenario 3** |  |  |
| *RSNR* | 0 | 0.25 | 0.5 | 0.75 | 1 |
| RR | 3.00 (0.15) | 5.08 (0.29) | 7.27 (0.51) | 8.20 (0.50) | 8.14 (0.57) |
| EN | 2.99 (0.16) | 5.08 (0.32) | 6.69 (0.55) | 6.43 (0.53) | 6.07 (0.36) |
| Lasso | 2.99 (0.17) | 5.39 (0.29) | 6.76 (0.53) | 6.64 (0.51) | 6.41 (0.38) |
| SREG | **2.90** (0.20) | **3.79** (0.28) | 4.18 (0.30) | 4.24 (0.29) | 4.32 (0.28) |
| FLR | 2.98 (0.15) | 4.10 (0.20) | 4.53 (0.29) | 4.45 (0.25) | 4.33 (0.24) |
| SVSIR | 3.48 (0.23) | 4.14 (0.29) | **3.61** (0.45) | **2.42** (0.21) | **2.02** (0.15) |

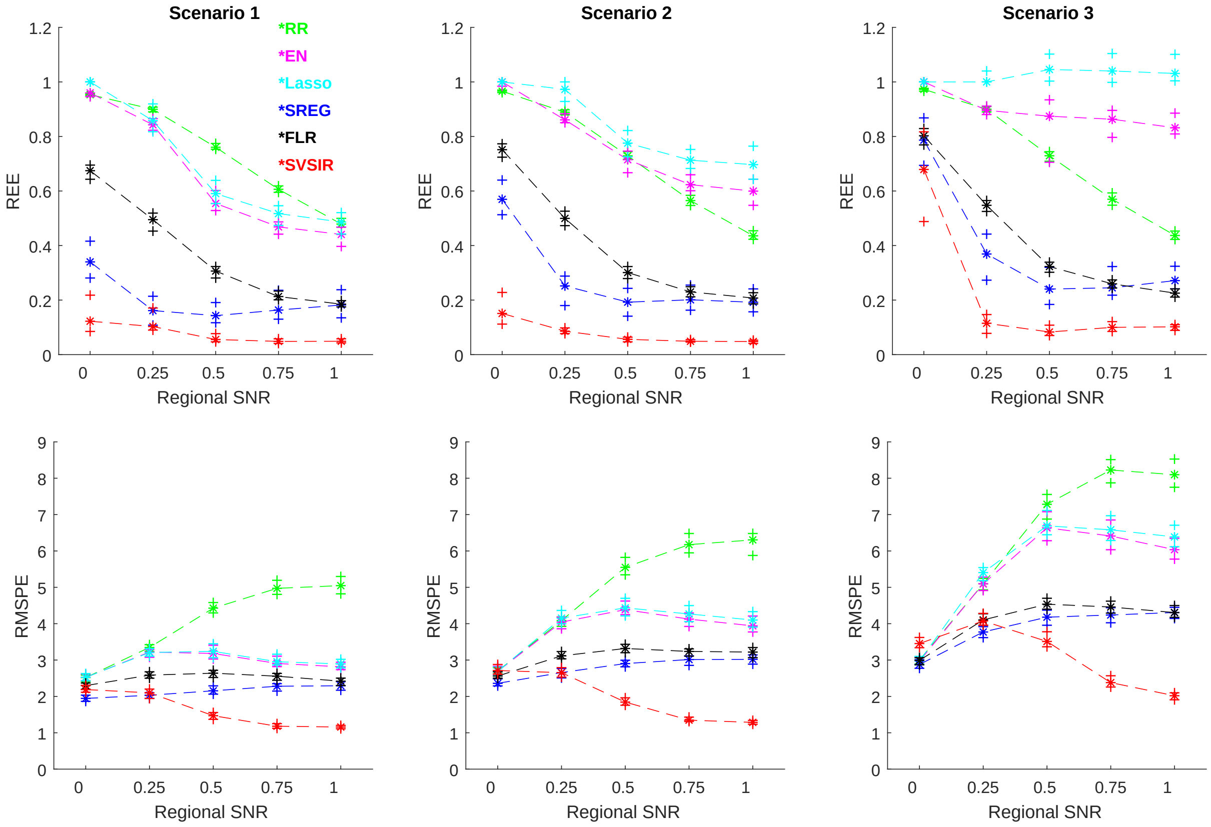


**Figure 3.7:** The box plot for the simulation studies. The plots in the top row are the estimation results for Scenarios 1 to 3 respectively. The plots in the bottom row are the prediction results. The asterisk signs represent the medians from 50 iterations, and the cross signs denote the 25 and 75 percentiles.

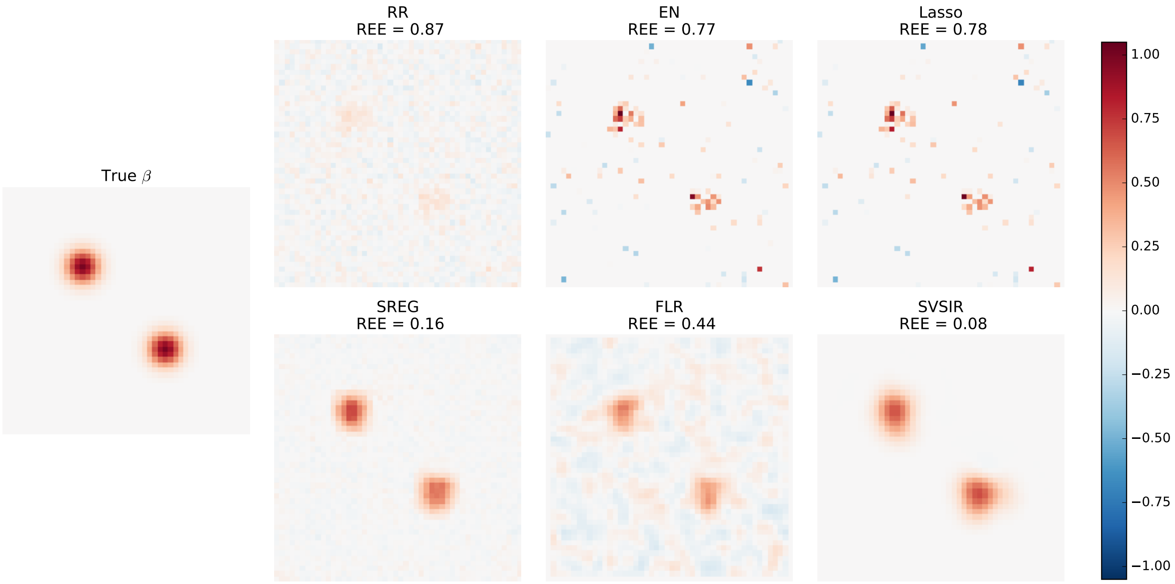


**Figure 3.8:** The plot of estimated coefficients in Scenario 1 with $RSNR = 0.25$. The image on the most left is the true coefficient. The top row is the estimation from ridge regression (RR), Elastic-Net (EN) and Lasso, and the bottom row is the estimation from spatial regularized regression (SREG), functional linear regression (FLR) and the proposed subject variant scalar-on-image regression (SVSIR). The relative estimation error (REE) for these methods are 0.87, 0.77, 0.78, 0.16, 0.44 and 0.08 respectively.

thus yield biased estimation. As the regional SNR gets larger, the estimation are improved for all six methods. For the SVSIR, it directly utilizes the regional signals and the heterogeneity, the increment of the estimation accuracy is more significant than other methods.

We evaluate the prediction accuracy on the independent test sets. The homogeneous models deliver overall under-performed predictions. This is because they yield biased estimation, and thus worse prediction. Furthermore, they apply the unified coefficients for each subjects while they are generated as heterogeneous samples. The propose SVSIR detects the active and inactive patterns in the training data, and use them to identify the active regions for the test samples through the pattern matching procedure defined in Equation (3.19). Under the moderate and strong regional SNR settings, its prediction accuracy is significantly improved. Under the homogeneous setting, i.e., $SNR = 0$, although SVSIR may yield a better estimation of the population disease map $\boldsymbol{\beta}$, its prediction is no better than those homogeneous models or even worse, mainly because of insufficient information to identify the active regions.

### 3.6 Real application

We apply the proposed SVSIR model in the Alzheimer's Disease Neuroimaging Initiative (ADNI) study. The ADNI study was launched in 2003 by the National Institute on Aging (NIA), the National Institute of Biomedical Imaging and Bioengineering (NIBIB), the Food and Drug Administration (FDA), private pharmaceutical companies and non-profit organizations. It collects magnetic resonance imaging (MRI) and positron emission tomography (PET) images, Cerebrospinal fluid (CSF), and blood biomarkers, among many others, to test whether those biological markers and neuropsychological assessments can be combined to measure the progression of MCI and early AD. More information about this study can be found at the ADNI website (http://www.adni-info.org/).

We aimed to utilize the T1-weighted MRI images collected at baseline to predict the cognitive scores of patients, which include MiniMental State Examination (MMSE) scores, Alzheimer's Disease Assessment Scale-cognitive score with 11 test items (ADAS-cog11) and 13 items (ADAS-cog13). The MMSE is a brief 30-point questionnaire test that is used to test or evaluate the cognitive impairment. It can be used to examine patient's arithmetic, memory and orientation. Generally, any score greater than or equal to 27 points (out of 30) indicates a normal cognition. Below this, MMSE score can indicate severe ($\leq$9 points), moderate (10-18 points) or mild (19-24 points) cognitive impairment ((Mungas, 1991)). The ADAS-Cog scores are also important to evaluate the stage of AD pathology and predict future progression. The scores are collected through a cognitive testing instrument in clinical trials. The test with 11 items has a total of 70 points and the one with 13 item with 85 points. Higher values indicate server disease progression. Fore more details, please refer to (Rosen et al., 1984) and (Petersen et al., 2005).

At present, structural MRI is one of the most popular and powerful imaging techniques for the diagnosis of AD. It is very interesting to use MRI data to predict the cognitive scores which can be used to diagnose the current disease status of AD. We exclude the subjects with missing measurements or low image quality, and select 749 participants in the ADNI study, including 165 AD, 366 MCI and 218 health control in our analysis. The demographical information of the subjects is summarized in Table 3.3.

**Table 3.3:** The demographical information of all subjects in data analysis. The mean values are reported, with standard deviations in parentheses.

| Diagnosis | Male | Female | Age | MMSE | ADAS-cog11 | ADAS-cog13 |
|---|---|---|---|---|---|---|
| AD | 86 | 79 | 75.47(7.33) | 23.32(1.98) | 18.32(6.02) | 28.65(7.44) |
| MCI | 238 | 128 | 74.78(7.23) | 26.98(1.78) | 11.55(4.48) | 18.66(6.32) |
| NC | 115 | 103 | 75.90(5.04) | 29.10(1.01) | 6.19(2.96) | 9.44(4.22) |

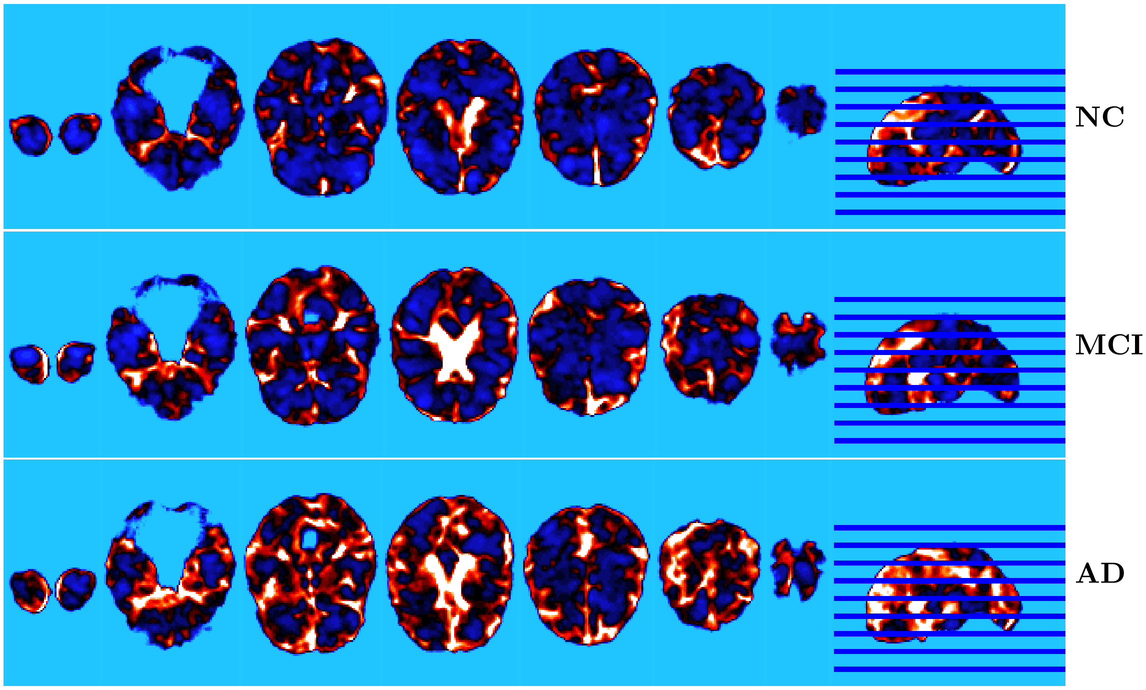


**Figure 3.9:** Three typical covariate images from patients with diagnostic labels as NC, MCI and AD respectively (from top to bottom).

All the MRI images are preprocessed by with anterior commissure and posterior commissure correction, N2 bias field correction, skull-stripping, intensity inhomogeneity correction, cerebellum removal, segmentation, and registration. We generate RAVENS-maps of the whole brain for each subject, using the deformation field obtained during registration (Davatzikos et al., 2001) and eventually obtain 749 images of size 128 × 128 × 128. Some typical example of each disease type are given in Figure 3.9.

We applied a training test splitting scheme to train and evaluate each model. In particular, we applied a stratified sampling on the whole dataset according to their diagnostic results and split it into training (80%) and test set (20%). We use the training set to fit the model, and evaluate it on the test set. For the methods requires a tuning parameter selection procedure, we adopted a inner

5-fold cross validation on the training set to select the model. We repeat the whole procedure 30 time, i.e., 30 independent random splits, and report the means and box plots of the results.

To predict each type of the cognitive scores, the six linear models mentioned in Section 3.5 are fitted based on image covariates. In order to compare the results from different cognitive measurements, we standardize the observed and estimated responses using the means and standard deviations calculated from the response values in the training set, and computed the RMSPE in Equation 3.20 based the rescaled values. Besides, we also evaluate the correlation between the predicted responses and the observed values, i.e.,

$$CORR = corr(\hat{Y}, Y).$$

The results are summarized in Tables 3.4 and 3.5, the box plots are given in Figure 3.10, and the coefficient images from the first iteration with responses as MMSE, ADAS-cog11 and ADAS-cog13 are displayed in Figure 3.11, 3.12 and 3.13 respectively. The prediction results with three different responses are consistent across all six models. The ridge regression achieves a overall better prediction error (around 0.90), while its performance in terms of correlation is not as good as other spatial methods. The sparse methods, Elastic Net and Lasso, yield high prediction errors and low correlations. This is mainly due to its inconsistency in variable selections and ultra sparsity in their estimated coefficient images. The spatial regularized regression achieves low prediction accuracies but high correlations. This is mainly explained by the underlying spatial structure in its coefficient images. The functional linear regression deliver a moderate prediction error as well as a moderate correlation. This is mainly due to the spatial smoothness induced by the Gaussian kernel in their coefficient images. The proposed SVSIR achieves either the best or the second best prediction error and the highest correlation among all six methods. This illustrates the advantage of modeling the heterogeneity. The spatial regularized methods yield overall higher correlations, which indicate their capability to identify disease-related regions in the estimated coefficients. The box plots in Figure 3.10 deliver the similar result in terms of RMSPE and correlation. Note that the Elastic Net and SVSIR methods have more variation in different iterations. For Elastic Net, it is mainly due to the inconsistent variable selection and the complexity in the tuning procedure. For the SVSIR, this variation mainly arise from the region estimation and assignments.

From the plots in Figures 3.11, 3.12 and 3.13, we can clearly see consistent patterns across the coefficient images from ridge regression, SREG, FLR ans SVSIR. The sparse methods Elastic Net and Lasso deliver ultra-sparse disease maps that are difficult to interpret biologically. The ridge regression and SREG yield spatial clustered estimation which can roughly capture the disease-related regions. The FLR delivers a smoother coefficient image with larger patchy signals than Ridge and SREG. The proposed SVSIR not only captures the spatial smoothing signals, but also yields moderate sparsity in the disease maps. This is mainly due to the benefit of using both the Gaussian kernel and the Potts prior.

**Table 3.4:** The mean value of the Root Mean Square Prediction Error (RMSPE) of the ADNI data analysis among 30 random splits. The smallest values are displayed in bold font and underlined, and the second smallest values are shown in bold font only.

| | MMSE | ADAS-cog11 | ADAS-cog13 |
|---|---|---|---|
| Ridge | **0.93** | $\underline{\mathbf{0.92}}$ | $\underline{\mathbf{0.89}}$ |
| Elastic | 1.13 | 1.16 | 1.21 |
| Lasso | 0.99 | 1.00 | 0.96 |
| SREG | 1.14 | 1.12 | 1.12 |
| FLR | 0.98 | 1.02 | 0.97 |
| SVSIR | $\underline{\mathbf{0.89}}$ | **0.93** | **0.93** |

**Table 3.5:** The mean value of the correlation between predicted scores and observed scores of the ADNI data analysis among 30 random splits. The largest values are displayed in bold font and underlined, and the second largest values are shown in bold font only.

| | MMSE | ADAS-cog11 | ADAS-cog13 |
|---|---|---|---|
| Ridge | 0.66 | 0.66 | 0.70 |
| Elastic | 0.53 | 0.51 | 0.50 |
| Lasso | 0.61 | 0.59 | 0.64 |
| SREG | **0.68** | **0.67** | **0.71** |
| FLR | 0.64 | 0.60 | 0.65 |
| SVSIR | $\underline{\mathbf{0.71}}$ | $\underline{\mathbf{0.70}}$ | $\underline{\mathbf{0.72}}$ |

Comparing the coefficients in Figure 3.11 with those in Figure 3.12 and 3.13, we can see that for the effect of the signals are opposite to each other. This is because the MMSE has a negative correlation with the disease severity, while the ADAS-cog scores has a positive correlation.

We overlay all the coefficient images obtained in this study on the Montreal Neurological Institute (MNI)-152 template (Fonov et al., 2011). Several regions are identified to as significant disease-related in the disease maps, such as the mid frontal gyrus, hippocampus, and temporal

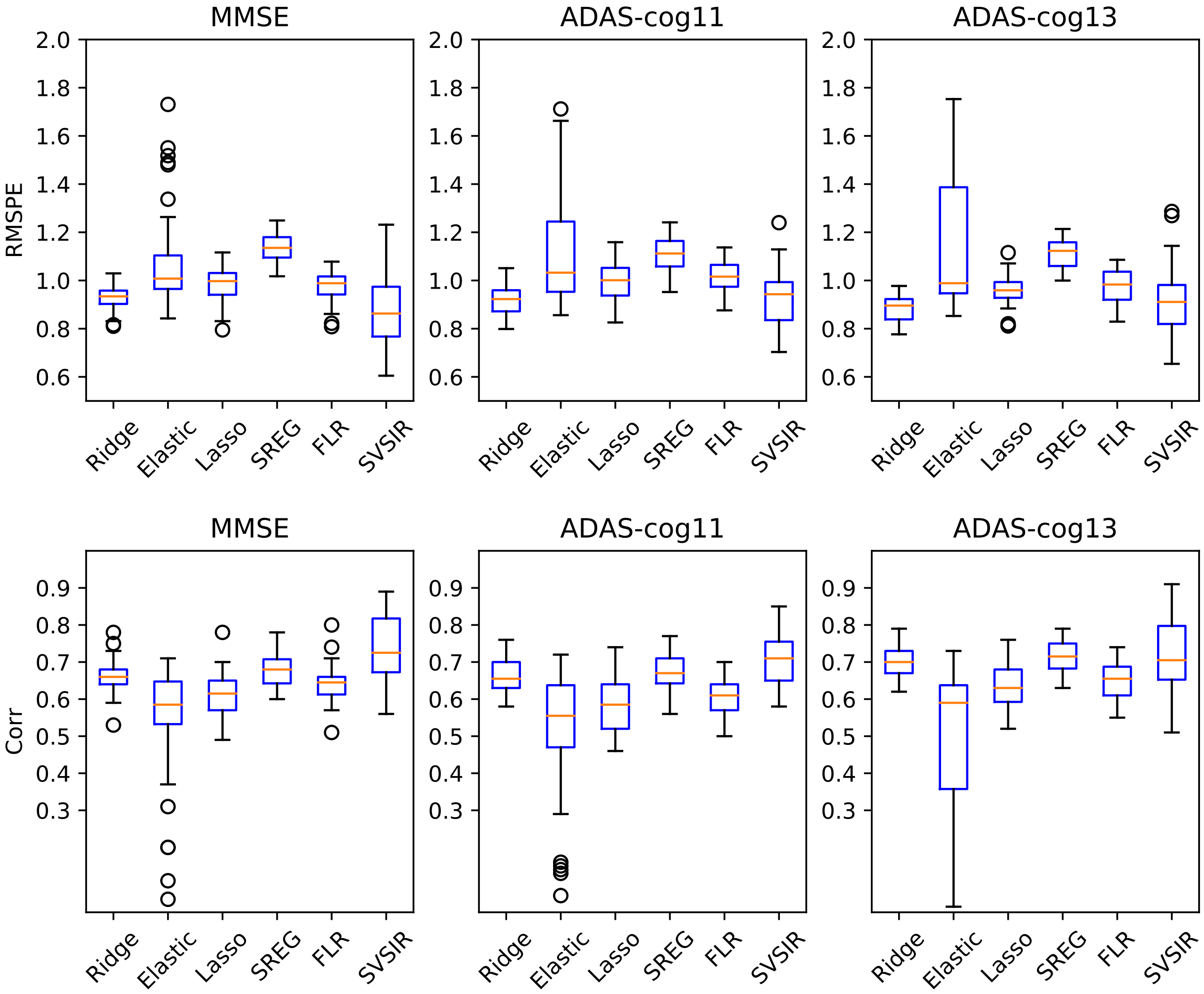


**Figure 3.10:** The box plot of the Root Mean Square Prediction Error (RMSPE) and the correlation between predicted scores and observed scores in the ADNI data analysis among 30 random splits.

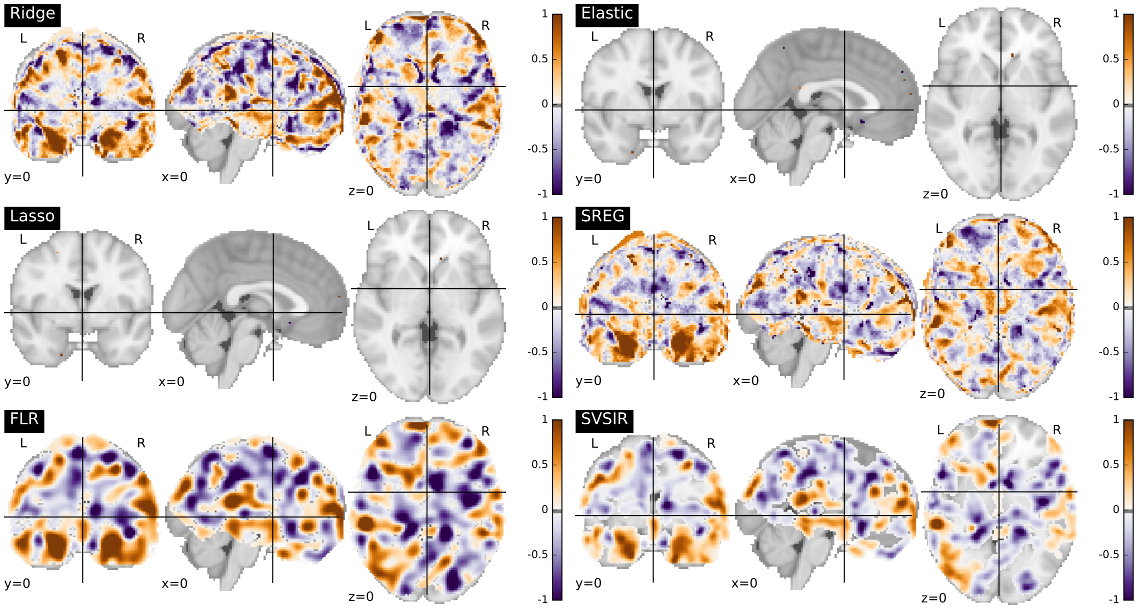


**Figure 3.11:** The plot of the coefficient images from all six methods, with MMSE as responses.

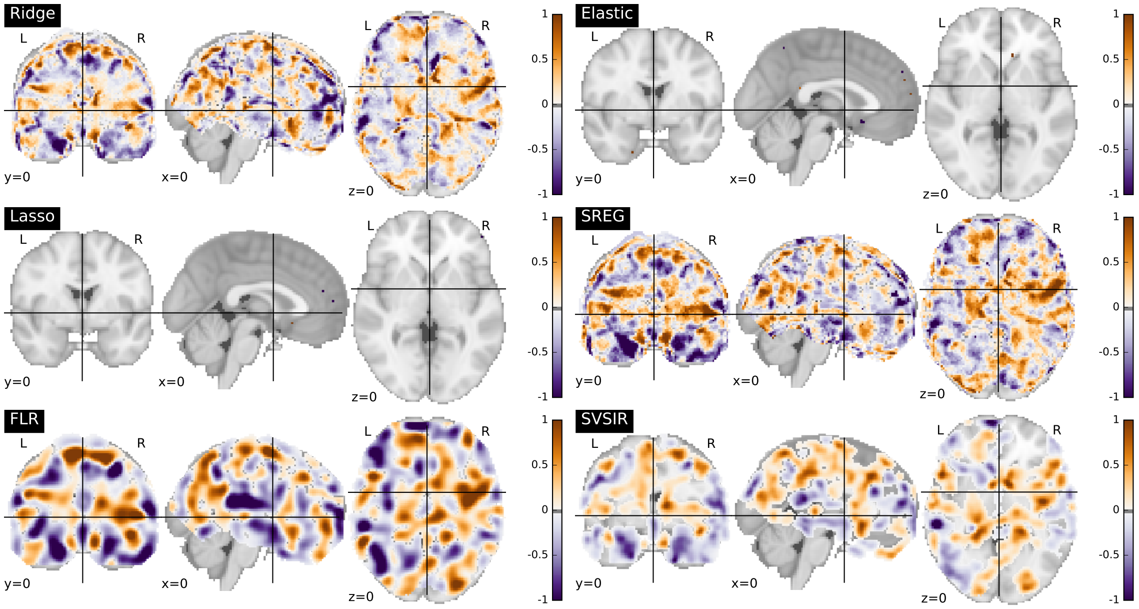


**Figure 3.12:** The plot of the coefficient images from all six methods, with ADAS-cog11 as responses.

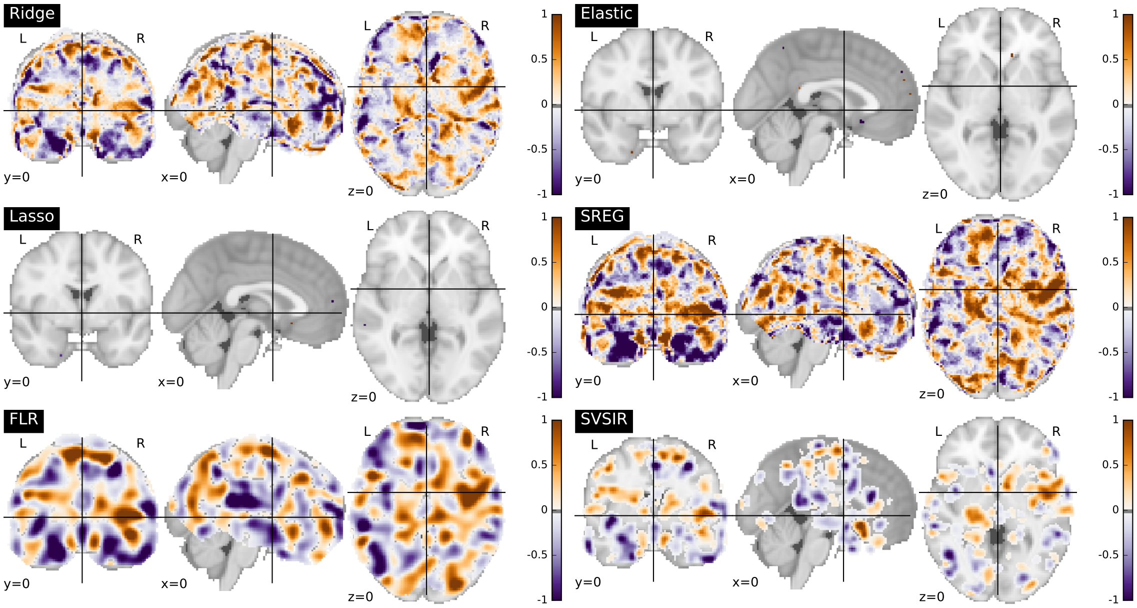


**Figure 3.13:** The plot of the coefficient images from all six methods, with ADAS-cog13 as responses.

gyrus. These regions have been proved in many research papers to be potentially related to the development of MCI and AD or functionally related to the cognitive ability. For instance, the superior and middle frontal gyrus are related to the logic thinking and planning, are progressively damaged during the disease development (Hirono et al., 1998). And the medial temporal lobe atrophy mainly appears in early stages of the Alzheimer's disease while generalized temporal lobe and global cerebral atrophy are characteristic of advanced AD (Killiany and Albert, 1993; Chan et al., 2001).

## 3.7 Discussion

In this chapter, we propose a novel subject variant scalar-on-image regression (SVSIR) model for the heterogeneous imaging data. The proposed SVSIR model utilizes the Gaussian reproducing kernel and preserves the spatial smoothness of the biological structure during the estimation. We introduce a hidden masking image with Potts prior that helps to enhance the spatial smoothness and rule out the irrelevant regions in the estimation. Our population-level coefficients capture the homogeneous regression relationship between clinical responses and the covariate images, and the individual-level coefficients utilize the heterogeneity structure and yield different patterns for active and inactive groups. The prediction performance is improved by incorporating heterogeneity structure among subjects.

## 3.8 Proofs

In this section, we provide the technical proofs for the theoretical results in Section 3.4.

### 3.8.1 Proof of Theorem 1

For simplicity, we assume $\tilde{\boldsymbol{X}}_i(t) = 0$ for all $t \in \mathcal{D}$ and $i \in [n]$. Let $l_n(\boldsymbol{f})$ denote the empirical loss and $C_n = \frac{1}{n}\sum_{i=1}^{n} \tilde{\boldsymbol{X}}_i(s)\tilde{\boldsymbol{X}}_i(t)$ represent the sample covariance operator, we then have

$$\begin{aligned}
l_n(\boldsymbol{f}) &= \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \int_{t\in\mathcal{D}} \tilde{\boldsymbol{X}}_i(t)\boldsymbol{f}(t)dt\right)^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}\left(\int_{t\in\mathcal{D}} \tilde{\boldsymbol{X}}_i(t)\left(\boldsymbol{\beta}(t) - \boldsymbol{f}(t)\right)dt + \epsilon_i\right)^2 \\
&= \|\boldsymbol{\beta} - \boldsymbol{f}\|_{C_n}^2 + \frac{1}{n}\sum_{i=1}^{n}\epsilon_i^2 + \frac{1}{n}\sum_{i=1}^{n}\epsilon_i\left(\int_{t\in\mathcal{D}} \tilde{\boldsymbol{X}}_i(t)\left(\boldsymbol{\beta}(t) - \boldsymbol{f}(t)\right)dt\right)
\end{aligned}$$

The associated expectation is given by

$$\begin{aligned}
l_\infty(\boldsymbol{f}) &= \mathbb{E}\left(Y - \int_{t\in\mathcal{D}} \tilde{\boldsymbol{X}}(t)\boldsymbol{f}(t)dt\right)^2 \\
&= \mathbb{E}\left(\int_{t\in\mathcal{D}} \tilde{\boldsymbol{X}}(t)\boldsymbol{\beta}(t) + \epsilon - \int_{t\in\mathcal{D}} \tilde{\boldsymbol{X}}(t)\boldsymbol{f}(t)dt\right)^2 \\
&= \mathbb{E}\left(\int_{t\in\mathcal{D}} \tilde{\boldsymbol{X}}(t)\left(\boldsymbol{\beta}(t) - \boldsymbol{f}(t)\right)dt\right)^2 + \sigma^2 \\
&= \|\boldsymbol{f} - \boldsymbol{\beta}\|_C^2 + \sigma^2
\end{aligned} \tag{3.21}$$

Let $l_{n\lambda} = l_n + \lambda J(\boldsymbol{f})$ and $l_{\infty\lambda} = l_\infty + \lambda J(\boldsymbol{f})$ denote the penalized loss functions, we can then write

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{f}\in\mathcal{H}}{\operatorname{argmin}}\left\{l_n(\boldsymbol{f}) + \lambda J(\boldsymbol{f})\right\} = \underset{\boldsymbol{f}\in\mathcal{H}}{\operatorname{argmin}}\left\{l_{n\lambda}(\boldsymbol{f})\right\}, \tag{3.22}$$

$$\bar{\boldsymbol{\beta}} = \underset{\boldsymbol{f}\in\mathcal{H}}{\operatorname{argmin}}\left\{l_\infty(\boldsymbol{f}) + \lambda J(\boldsymbol{f})\right\} = \underset{\boldsymbol{f}\in\mathcal{H}}{\operatorname{argmin}}\left\{l_{\infty\lambda}(\boldsymbol{f})\right\}. \tag{3.23}$$

Clearly, $\hat{\boldsymbol{\beta}} - \boldsymbol{\beta} = \left(\hat{\boldsymbol{\beta}} - \bar{\boldsymbol{\beta}}\right) + \left(\bar{\boldsymbol{\beta}} - \boldsymbol{\beta}\right)$. Note that, the first term on the right hand side (RHS) is associated with the estimation procedure, thus is denoted as the estimation error, and the second term is denoted as the model error. By the triangle inequality, it suffices to establish an unified bound for the two terms. We will establish the bound in Lemmas 1 and 2, which will complete the proof of Theorem 1.

**Lemma 1.** *Under assumptions A.1-A.5, the modeling error satisfies*

$$\|\bar{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_C^2 = O_p(n^{-\frac{2(a+b)}{2(a+b)+1}}),$$

*with* $\lambda \asymp n^{-\frac{2(a+b)}{2(a+b)+1}}$.

**Lemma 2.** *Under assumptions A.1-A.5, the modeling error satisfies*

$$\|\hat{\boldsymbol{\beta}} - \bar{\boldsymbol{\beta}}\|_C^2 = O_p(n^{-\frac{2(a+b)}{2(a+b)+1}}),$$

*with* $\lambda \asymp n^{-\frac{2(a+b)}{2(a+b)+1}}$.

The proof of the Lemmas 1 and 2 are based on simultaneous decomposition of the norms associated with $R$ and $C$. Recall that assumption A.5 indicates $\|\boldsymbol{f}\|_R^2 = \|\boldsymbol{f}\|_C^2 + \lambda J(\boldsymbol{f})$, and $R^{1/2}CR^{1/2}$ can be decomposed with $\nu_k$ and $\xi_k$. The simultaneous decomposition is given in Lemma 3.

**Lemma 3.** *Under assumptions A.3 and A.5, the norms associated with $R$ and $C$ can be simultaneously decomposed with the bases $\left\{\omega_k = \nu_k^{-1/2} R^{1/2} \xi_k,\, k \geq 1\right\}$, i.e., for any $\boldsymbol{f} \in \mathcal{H}$, we have $\boldsymbol{f} = \sum_{k=1}^{\infty} f_k \omega_k$, where $f_k = v_k \langle \boldsymbol{f}, \omega_k \rangle_R$. Furthermore,*

$$\|\boldsymbol{f}\|_R^2 = \sum_{k=1}^{\infty} \left(1 + \frac{\lambda}{\lambda_k^*}\right) f_k^2, \tag{3.24}$$

$$\|\boldsymbol{f}\|_C^2 = \sum_{k=1}^{\infty} f_k^2, \tag{3.25}$$

$$J(\boldsymbol{f}) = \frac{1}{\lambda}\left(\|\boldsymbol{f}\|_R^2 - \|\boldsymbol{f}\|_C^2\right) = \sum_{k=1}^{\infty} \frac{f_k^2}{\lambda_k^*}. \tag{3.26}$$

Note that the eigen functions $\xi_k$'s and $\omega_k$'s are difficulty to construct in general. The bounds of the norms are established through the decay rate of the eigenvalues. The proof of the lemma can be found in (Yuan et al., 2010).

We first establish the bound for modeling error:

*Proof of Lemma 1:* Let $\boldsymbol{\beta} = \sum_{k=1}^{\infty} b_k \omega_k$, $\hat{\boldsymbol{\beta}} = \sum_{k=1}^{\infty} \hat{b}_k \omega_k$ and $\bar{\boldsymbol{\beta}} = \sum_{k=1}^{\infty} \bar{b}_k \omega_k$ be the eigen decomposition of the three functions respectively. By Equations (3.21) and (3.25) we have

$$l_\infty(\boldsymbol{f}) = \sigma^2 + \sum_{k=1}^{\infty} (f_k - b_k)^2 .$$

Thus,

$$\begin{aligned} l_\infty(\boldsymbol{f}) + \lambda J(\boldsymbol{f}) &= \sum_{k=1}^{\infty} (f_k - b_k)^2 + \lambda J(\boldsymbol{f}) \\ &= \sum_{k=1}^{\infty} \left( (f_k - b_k)^2 + \frac{\lambda f_k^2}{\lambda_k^*} \right). && \text{by Equation (3.26)} \end{aligned}$$

By Equation 3.23, and the decomposition of $\bar{\boldsymbol{\beta}}$, we have

$$\begin{aligned} \bar{b}_k &= \operatorname*{argmin}_{f_k} \left\{ (f_k - b_k)^2 + \frac{\lambda f_k^2}{\lambda_k^*} \right\} \\ &= \frac{b_k}{1 + \lambda/\lambda_k^*}. \end{aligned}$$

Therefore, we have

$$\begin{aligned} \|\bar{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_C^2 &= \sum_{k=1}^{\infty} \left(b_k - \bar{b}_k\right)^2 \\ &= \sum_{k=1}^{\infty} \left( b_k - \frac{b_k}{1 + \lambda/\lambda_k^*} \right)^2 \\ &= \lambda^2 \sum_{k=1}^{\infty} \frac{b_k^2}{\lambda_k^*} \frac{\lambda_k^*}{\left(\lambda + \lambda_k^*\right)^2} \\ &\leq J(\boldsymbol{\beta}) \sup_k \frac{\lambda_k^*}{\left(\lambda + \lambda_k^*\right)^2} && \text{by Equation (3.26)} \\ &\leq J(\boldsymbol{\beta}) \sup_{x>0} \frac{1}{\left(\lambda/\sqrt{x} + \sqrt{x}\right)^2} \end{aligned}$$

$$\leq J(\boldsymbol{\beta})\frac{\lambda}{4} \qquad\qquad \text{by letting } x = \sqrt{\lambda}.$$

Thus we can conclude that, given $J(\boldsymbol{\beta})$ is bounded from above,

$$\|\bar{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_C^2 = O_p(\lambda). \tag{3.27}$$

Since $\lambda \asymp n^{-\frac{2(a+b)}{2(a+b)+1}}$, we have

$$\|\bar{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_C^2 = O_p(n^{-\frac{2(a+b)}{2(a+b)+1}}). \tag{3.28}$$

□

Now we establish a bound for the estimation error.

*Proof of Lemma 2:* We introduce a intermediate variable $\tilde{\boldsymbol{\beta}}$ between $\hat{\boldsymbol{\beta}}$ and $\bar{\boldsymbol{\beta}}$, and bound the modeling error in terms of $\|\hat{\boldsymbol{\beta}} - \tilde{\boldsymbol{\beta}}\|_C^2$ and $\|\bar{\boldsymbol{\beta}} - \tilde{\boldsymbol{\beta}}\|_C^2$ since $\hat{\boldsymbol{\beta}} - \bar{\boldsymbol{\beta}} = (\hat{\boldsymbol{\beta}} - \tilde{\boldsymbol{\beta}}) + (\bar{\boldsymbol{\beta}} - \tilde{\boldsymbol{\beta}})$.

The variable $\tilde{\boldsymbol{\beta}}$ is then constructed by as follows,

$$\tilde{\boldsymbol{\beta}} = \bar{\boldsymbol{\beta}} - (l''_{\infty\lambda})^{-1} l'_{n\lambda}(\bar{\boldsymbol{\beta}}), \tag{3.29}$$

where $l'$ and $l''$ denote the first and second derivatives of the functional $l$.

Due to the optimality of $\bar{\boldsymbol{\beta}}$, we have $l'_{\infty\lambda}(\bar{\boldsymbol{\beta}}) = 0$, which further implies that

$$l'_{n\lambda}(\bar{\boldsymbol{\beta}}) = l'_{n\lambda}(\bar{\boldsymbol{\beta}}) - l'_{\infty\lambda}(\bar{\boldsymbol{\beta}}) = l'_n(\bar{\boldsymbol{\beta}}) - l'_\infty(\bar{\boldsymbol{\beta}}).$$

Observe that

$$\begin{aligned}
l'_n(\boldsymbol{f}) &= -\frac{2}{n}\sum_{i=1}^{n}\left(\int_{t\in\mathcal{D}} \tilde{\boldsymbol{X}}_i(t)\left(\boldsymbol{\beta}(t) - \boldsymbol{f}(t)\right)dt + \epsilon_i\right)\tilde{\boldsymbol{X}}_i(s);\\
l''_n(\boldsymbol{f}) &= 2C_n(s,t);\\
l'_\infty(\boldsymbol{f}) &= -2\mathbb{E}\left(\int_{t\in\mathcal{D}} \tilde{\boldsymbol{X}}(t)\left(\boldsymbol{\beta}(t) - \boldsymbol{f}(t)\right)dt\right)\tilde{\boldsymbol{X}}(s);\\
l''_\infty(\boldsymbol{f}) &= 2C(s,t)
\end{aligned}$$

For $k \geq 1$, we have

$$\begin{aligned}
&\mathbb{E}\left[l'_{n\lambda}(\bar{\boldsymbol{\beta}})\omega_k\right]^2 \\
=&\mathbb{E}\left[l'_n(\bar{\boldsymbol{\beta}})\omega_k - l'_\infty(\bar{\boldsymbol{\beta}})\omega_k\right]^2 \\
=&\mathbb{E}\left[-\frac{2}{n}\sum_{i=1}^{n}\left(\int_{t\in\mathcal{D}}\tilde{\boldsymbol{X}}_i(t)\left(\boldsymbol{\beta}(t)-\bar{\boldsymbol{\beta}}(t)\right)dt+\epsilon_i\right)\int_{s\in\mathcal{D}}\tilde{\boldsymbol{X}}_i(s)\omega_k(s)ds\right. \\
&\left.-2\mathbb{E}\left(\int_{t\in\mathcal{D}}\tilde{\boldsymbol{X}}(t)\left(\boldsymbol{\beta}(t)-\bar{\boldsymbol{\beta}}(t)\right)dt\right)\int_{s\in\mathcal{D}}\tilde{\boldsymbol{X}}(s)\omega_k(s)ds\right]^2 \\
=&\frac{4}{n}Var\left[\left(\int_{t\in\mathcal{D}}\tilde{\boldsymbol{X}}(t)\left(\boldsymbol{\beta}(t)-\bar{\boldsymbol{\beta}}(t)\right)dt+\epsilon\right)\int_{s\in\mathcal{D}}\tilde{\boldsymbol{X}}(s)\omega_k(s)ds\right] \\
\leq&\frac{4}{n}\left\{\mathbb{E}\left[\int_{t\in\mathcal{D}}\tilde{\boldsymbol{X}}(t)\left(\boldsymbol{\beta}(t)-\bar{\boldsymbol{\beta}}(t)\right)dt\int_{s\in\mathcal{D}}\tilde{\boldsymbol{X}}(s)\omega_k(s)ds\right]^2+\mathbb{E}\left[\epsilon^2\right]\mathbb{E}\left[\int_{s\in\mathcal{D}}\tilde{\boldsymbol{X}}(s)\omega_k(s)ds\right]^2\right\}.
\end{aligned}$$

Now we are in the position of bounding the two terms in the RHS.

$$\begin{aligned}
&\mathbb{E}\left[\int_{t\in\mathcal{D}}\tilde{\boldsymbol{X}}(t)\left(\boldsymbol{\beta}(t)-\bar{\boldsymbol{\beta}}(t)\right)dt\int_{s\in\mathcal{D}}\tilde{\boldsymbol{X}}(s)\omega_k(s)ds\right]^2 \\
\leq&\left\{\mathbb{E}\left[\int_{t\in\mathcal{D}}\tilde{\boldsymbol{X}}(t)\left(\boldsymbol{\beta}(t)-\bar{\boldsymbol{\beta}}(t)\right)dt\right]^4\times\mathbb{E}\left[\int_{s\in\mathcal{D}}\tilde{\boldsymbol{X}}(s)\omega_k(s)ds\right]^4\right\}^{1/2} && \text{by Cauchi-Schwarz inequality} \\
\leq&2\mathbb{E}\left[\int_{t\in\mathcal{D}}\tilde{\boldsymbol{X}}(t)\left(\boldsymbol{\beta}(t)-\bar{\boldsymbol{\beta}}(t)\right)dt\right]^2\times\mathbb{E}\left[\int_{s\in\mathcal{D}}\tilde{\boldsymbol{X}}(s)\omega_k(s)ds\right]^2 && \text{by the Gassian assumption in A.1} \\
=&2\|\boldsymbol{\beta}-\bar{\boldsymbol{\beta}}\|_C^2\times\|\omega_k\|_C^2 \\
=&2\|\boldsymbol{\beta}-\bar{\boldsymbol{\beta}}\|_C^2 && \|\omega_k\|_C^2=1 \text{ by Equation (3.25)}
\end{aligned}$$

Therefore, we have

$$\mathbb{E}\left[l'_{n\lambda}(\bar{\boldsymbol{\beta}})\omega_k\right]^2 \leq \frac{4}{n}\left(O_p(\lambda)+\sigma^2\right) \leq C_0 n^{-1},$$

given that $\lambda$ is bounded from above and $C_0$ represents a big constant.

We establish the bound for $\|\tilde{\boldsymbol{\beta}}-\bar{\boldsymbol{\beta}}\|_C^2$ now, i.e,

$$\begin{aligned}
&\mathbb{E}\|\tilde{\boldsymbol{\beta}}-\bar{\boldsymbol{\beta}}\|_C^2 \\
=&\mathbb{E}\|(l''_{\infty\lambda})^{-1}l'_{n\lambda}(\bar{\boldsymbol{\beta}})\|_C^2
\end{aligned}$$

$$
\begin{aligned}
=&\frac{1}{4}\mathbb{E}\left[\sum_{k=1}^{\infty}\left(1+\frac{\lambda}{\lambda_k^*}\right)^{-2}\left(l'_{n\lambda}(\bar{\boldsymbol{\beta}})\omega_k\right)^2\right] \\
\leq& C_0 n^{-1}\sum_{k=1}^{\infty}\left(1+\frac{\lambda}{\lambda_k^*}\right)^{-2} \\
\asymp& C_0 n^{-1}\sum_{k=1}^{\infty}\left(1+\lambda k^{2(a+b)}\right)^{-2} && \text{by assumption A.5} \\
\asymp& C_0 n^{-1}\int_{x>1}\left(1+\lambda x^{2(a+b)}\right)^{-2}dx \\
=& C_0 n^{-1}\lambda^{-\frac{1}{2(a+b)}}\int_{t>\lambda^{\frac{1}{2(a+b)}}}\left(1+t^{2(a+b)}\right)^{-2}dt && \text{by letting } t=\lambda^{\frac{1}{2(a+b)}}x \\
=& O_p(n^{-1}\lambda^{-\frac{1}{2(a+b)}})
\end{aligned}
$$

Given $\lambda \asymp n^{-\frac{2(a+b)}{2(a+b)+1}}$, we have

$$
\mathbb{E}\|\tilde{\boldsymbol{\beta}}-\bar{\boldsymbol{\beta}}\|_C^2 = O_p(n^{-\frac{2(a+b)}{2(a+b)+1}}). \tag{3.30}
$$

Since $l_{n\lambda}(\boldsymbol{f})$ is quadratic in $\boldsymbol{f}$, we have

$$
\begin{aligned}
&l'_{n\lambda}(\bar{\boldsymbol{\beta}}) + l''_{n\lambda}(\bar{\boldsymbol{\beta}})\left(\hat{\boldsymbol{\beta}}-\bar{\boldsymbol{\beta}}\right) = l'_{n\lambda}(\hat{\boldsymbol{\beta}}) = 0 \\
\Rightarrow & l'_{n\lambda}(\bar{\boldsymbol{\beta}}) = -l''_{n\lambda}(\bar{\boldsymbol{\beta}})\left(\hat{\boldsymbol{\beta}}-\bar{\boldsymbol{\beta}}\right) \qquad (3.31)
\end{aligned}
$$

Note that

$$
\begin{aligned}
& l''_{\infty\lambda}(\bar{\boldsymbol{\beta}})\left(\hat{\boldsymbol{\beta}}-\tilde{\boldsymbol{\beta}}\right) \\
=& l''_{\infty\lambda}(\bar{\boldsymbol{\beta}})\left(\left(\hat{\boldsymbol{\beta}}-\bar{\boldsymbol{\beta}}\right)-\left(\tilde{\boldsymbol{\beta}}-\bar{\boldsymbol{\beta}}\right)\right) \\
=& l''_{\infty\lambda}(\bar{\boldsymbol{\beta}})\left(\hat{\boldsymbol{\beta}}-\bar{\boldsymbol{\beta}}\right) - l''_{\infty\lambda}(\bar{\boldsymbol{\beta}})\left(-l''_{\infty\lambda}(\bar{\boldsymbol{\beta}})^{-1}l'_{n\lambda}(\bar{\boldsymbol{\beta}})\right) && \text{by Equation (3.29)} \\
=& l''_{\infty\lambda}(\bar{\boldsymbol{\beta}})\left(\hat{\boldsymbol{\beta}}-\bar{\boldsymbol{\beta}}\right) + l'_{n\lambda}(\bar{\boldsymbol{\beta}}) \\
=& l''_{\infty\lambda}(\bar{\boldsymbol{\beta}})\left(\hat{\boldsymbol{\beta}}-\bar{\boldsymbol{\beta}}\right) - l''_{n\lambda}(\bar{\boldsymbol{\beta}})\left(\hat{\boldsymbol{\beta}}-\bar{\boldsymbol{\beta}}\right) && \text{by Equation (3.31)} \\
=& l''_{\infty}(\bar{\boldsymbol{\beta}})\left(\hat{\boldsymbol{\beta}}-\bar{\boldsymbol{\beta}}\right) - l''_{n}(\bar{\boldsymbol{\beta}})\left(\hat{\boldsymbol{\beta}}-\bar{\boldsymbol{\beta}}\right).
\end{aligned}
$$

Thus we have

$$\hat{\boldsymbol{\beta}} - \tilde{\boldsymbol{\beta}} = l''_{\infty\lambda}(\bar{\boldsymbol{\beta}})^{-1}\left(l''_{\infty}(\bar{\boldsymbol{\beta}})\left(\hat{\boldsymbol{\beta}} - \bar{\boldsymbol{\beta}}\right) - l''_{n}(\bar{\boldsymbol{\beta}})\left(\hat{\boldsymbol{\beta}} - \bar{\boldsymbol{\beta}}\right)\right).$$

and

$$\begin{aligned}
&\mathbb{E}\|\hat{\boldsymbol{\beta}} - \tilde{\boldsymbol{\beta}}\|_C^2 \\
=&\mathbb{E}\|l''_{\infty\lambda}(\bar{\boldsymbol{\beta}})^{-1}\left(l''_{\infty}(\bar{\boldsymbol{\beta}})\left(\hat{\boldsymbol{\beta}} - \bar{\boldsymbol{\beta}}\right) - l''_{n}(\bar{\boldsymbol{\beta}})\left(\hat{\boldsymbol{\beta}} - \bar{\boldsymbol{\beta}}\right)\right)\|_C^2 \\
=&\frac{1}{4}\sum_{k=1}^{\infty}\left(1+\frac{\lambda}{\lambda_k^*}\right)^{-2}\sum_{j=1}^{\infty}\left(\hat{b}_j - \bar{b}_j\right)\mathbb{E}\iint_{s\in\mathcal{D},t\in\mathcal{D}}\omega_j(s)\left(C_n(s,t)-C(s,t)\right)\omega_k(t)dsdt \\
\leq&\frac{1}{4}\sum_{k=1}^{\infty}\left(1+\frac{\lambda}{\lambda_k^*}\right)^{-2}\sum_{j=1}^{\infty}\left(\hat{b}_j - \bar{b}_j\right)^2\sum_{j=1}^{\infty}\left(\mathbb{E}\iint_{s\in\mathcal{D},t\in\mathcal{D}}\omega_j(s)\left(C_n(s,t)-C(s,t)\right)\omega_k(t)dsdt\right)^2 \\
\asymp&O_p(\lambda^{\frac{1}{2(a+b)}})\mathbb{E}\|\hat{\boldsymbol{\beta}} - \bar{\boldsymbol{\beta}}\|_C^2\sum_{j=1}^{\infty}\frac{1}{n}Var\left[\iint_{s\in\mathcal{D},t\in\mathcal{D}}\omega_j(s)X(s)X(t)\omega_k(t)dsdt\right]
\end{aligned}$$

Note that

$$\begin{aligned}
&Var\left[\iint_{s\in\mathcal{D},t\in\mathcal{D}}\omega_j(s)X(s)X(t)\omega_k(t)dsdt\right] \\
\leq&\mathbb{E}\left[\int_{s\in\mathcal{D}}\omega_j(s)X(s)ds\int_{t\in\mathcal{D}}\omega_k(t)X(t)dt\right]^2 \\
\leq&\left\{\mathbb{E}\left[\int_{s\in\mathcal{D}}\omega_j(s)X(s)ds\right]^4\mathbb{E}\left[\int_{t\in\mathcal{D}}\omega_k(t)X(t)dt\right]^4\right\}^{1/2} \\
\asymp&O_p(1)
\end{aligned}$$

Therefore, we have

$$\mathbb{E}\|\hat{\boldsymbol{\beta}} - \tilde{\boldsymbol{\beta}}\|_C^2 \asymp O_p(n^{-1}\lambda^{\frac{1}{2(a+b)}}\mathbb{E}\|\hat{\boldsymbol{\beta}} - \bar{\boldsymbol{\beta}}\|_C^2) = o_p(\mathbb{E}\|\hat{\boldsymbol{\beta}} - \bar{\boldsymbol{\beta}}\|_C^2) \qquad (3.32)$$

Combining the results from Equations (3.30),(3.32), we have complete the proof. □

CHAPTER 4

# MCNN: Masking Convolutional Neural Network for Image Classification and Regression

## 4.1 Introduction

In the past decade, convolutional neural network (CNN) models have received much attention due to their competitive performance in various tasks, including object classification (Krizhevsky et al., 2012; Simonyan and Zisserman, 2014; He et al., 2016; Huang et al., 2017) and semantic segmentation (Ronneberger et al., 2015; Long et al., 2015; Badrinarayanan et al., 2017; Chen et al., 2018). In object classification problems, we may be interested in identifying objects in the images that are associated with class labels. Many CNN-based models facilitate learning data-driven, highly representative, layered hierarchical image features from complex datasets; an example is ImageNet (Deng et al., 2009). These models are quite robust to the large variation of object locations and sizes and tend to aggregate information over the whole image. However, it can be difficult to interpret most high-level image features extracted from CNNs, thus interpretability remains to be a major challenge (Zhang and Zhu, 2018).

To address this challenge, we propose to modify standard CNNs to achieve better model interpretability and prediction in certain supervised learning problems without any additional human supervision. Our proposed approach differs from that of most existing methods in the direction of understanding neural network representations, but can be regarded as a set of neural network models with interpretable/disentangled representations. See a comprehensive review of various methods for improving the models ability to interpret CNNs in (Zhang and Zhu, 2018).

The aim of this chapter is to propose a set of masked CNN (MCNN) models with high model interpretability and better prediction. The key components of MCNN are shown in Figure 4.1 and summarized as follows:

- Introduce a latent binary network to extract informative regions of interest (ROIs) for each image that contain informative signals for prediction.

- Simultaneously learn the latent binary network with CNN for achieving better prediction in various supervised learning problems.

Our MCNN can be regarded as a novel extension of the standard two-stage computer-aided diagnostic approach, which consists of segmenting objects of interest (e.g., a tumor) in the first stage and using segmented objects for prediction in the second stage. In contrast, MCNN is a simultaneous segmentation-prediction approach that integrates a semantic segmentation network and CNN into a single neural network model.

Compared with the existing methods represented in the literature, three major methodological contributions in this chapter are summarized as follows:

- First, MCNN carries out population semantic segmentation across all images based on the latent binary network. It focuses on objects that are highly predictive of the response of interest; whereas standard semantic segmentation networks are developed to identify ROIs in individual images that represent different objects. Moreover, although semantic segmentation networks are able to deliver pixel-wise annotations, they require extensive human labeling in preparing training samples, which are expensive to acquire (Chen et al., 2017; Yu and Koltun, 2015; Long et al., 2015; Ronneberger et al., 2015).

- Second, we propose to learn the latent binary network with CNN to improve its interpretability, which can be widely applicable to CNNs with different architectures.

- Third, since MCNN focuses on the informative object learned from the latent binary network by ruling out the irrelevant regions, it may enhance predictive signals, subsequently leading to better prediction.

The rest of this chapter is organized as follows. In Section 2, we discuss three different scenarios that MCNN mainly applies for. In Section 4.3, we introduce the technical details of the proposed MCNN. We demonstrate the performance of MCNN in two synthetic experiments in Section 4.4 and two real applications in Section 4.5. Section 4.6 concludes this chapter with some discussion.

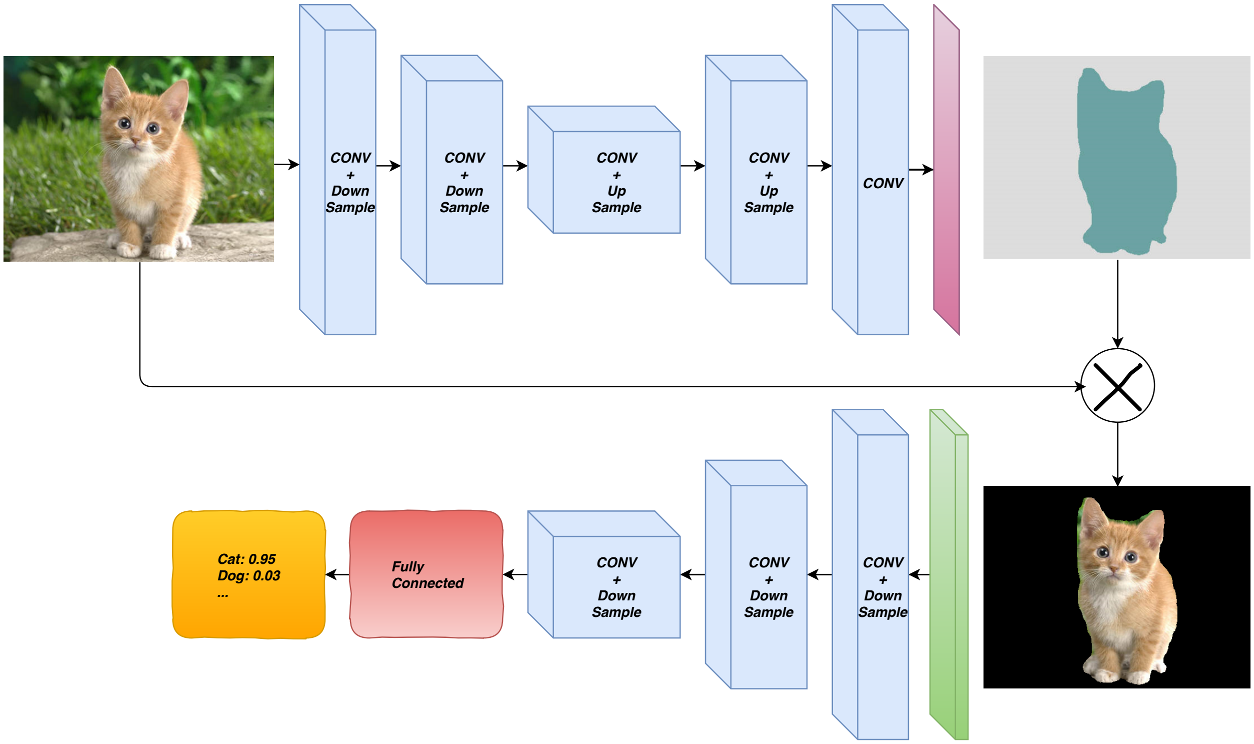


**Figure 4.1:** A representative structure of MCNN consisting of the latent binary mapping module with U-net and a classification network based on VGG (Simonyan and Zisserman, 2014). Here ⊗ denotes the element-wise product between the masking matrix and the input image.

## 4.2 Data structure

MCNN mainly solve the image based prediction problems corresponds to three different scenarios in real applications.

First, when precise pixel-wise annotations of input images are available, the network can conduct simultaneous segmentation and prediction, and utilize the structure of both networks to improve the overall performance.

Second, MCNN model can utilize the binary images that roughly capture the predictive regions, to improve the prediction and generate refined masking images based on the rough masks.

Third, in practice, especially in analysis of medical images, imaging data are usually mixed with annotated and unannotated samples, or mixed with high and low qualities annotations. The proposed MCNN efficiently handles this scenario by assigning various weights for different samples.

We denote these three cases as Scenario 1-3 in the rest of the chapter. Details about the strategies to handle each scenario are discussed in Section 4.3.3.

### 4.3 Masked convolutional neural network

MCNN consists of two connected modules: a segmentation module that estimates the latent mask maps and a prediction module to perform regression or classification. We denote the network architecture of MCNN as

$$\widehat{y} = \mathcal{F}_1 \left( X \otimes \mathcal{F}_0 \left( X \right) \right),$$

where $X$ represents the input image; $\mathcal{F}_0$ and $\mathcal{F}_1$ respectively correspond to the segmentation and prediction networks; $\widehat{y}$ denotes the predicted value, which can be a vector of $K$ probabilities for a $K$-category classification problem or real values in the regression setting; and "$\otimes$" represents the pixel-wise multiplication.

#### 4.3.1 Segmentation module

The segmentation module enhances the predictive signals by masking off 'background noises'. It estimates a latent binary map (or mask image) $\widehat{M}$ from the input image $X$ such that we have $\widehat{M} = \mathcal{F}_0(X)$. For instance, we set $F_0(\cdot)$ to be the U-net architecture that consists of a symmetric structure of auto-encoders and decoders (Ronneberger et al., 2015). In this case, the module utilizes convolutional and down-sampling layers to aggregate the information over the whole imaging space, gradually expands the feature maps by deconvolutional or up-sampling operations, and eventually constructs a probability map $\widehat{M}$ as the mask. **Compared with standard semantic segmentation methods, MCNN does not require pixel-wise annotated images, even though knowing such annotated images may substantially increase the discrimination power.** Furthermore, the individual mask images $\widehat{M}$ explicitly localize important ROIs that contribute most to prediction outputs at the pixel level. As shown in numerical examples, the individual masking images $\widehat{M}$ can efficiently handle objects with large variation in terms of both location and size across subjects. Thus, the use of $\widehat{M}$ dramatically improves the models ability to interpret deep neural networks.

### 4.3.2 Prediction module

The prediction module $\widehat{y} = \mathcal{F}_1(\widehat{M} \otimes X)$ is directly connected with the output end of the segmentation module. Specifically, it takes the masked images as input and estimates the classification probabilities or numerical response according to the type of learning problems. This module can adopt any CNN architecture. It sequentially processes the masked input images by using convolutional and down-sampling operations and utilizes fully connected layers to estimate the final predicted responses.

### 4.3.3 Loss functions

The loss function of MCNN is written as the weighted sum of a segmentation loss and a prediction loss:

$$l(\widehat{y}, \widehat{M}) = l_p(\widehat{y}) + \lambda l_s(\widehat{M}),$$

where $\lambda$ balances the two modules. Specifically, in Scenario 1, we have masking images with precise pixel-wise annotations, so we assign a relatively large $\lambda$ to emphasize the segmentation module. In Scenario 2 with "imprecise" pixel-wise annotations, we use small values for $\lambda$. In the case of mixed samples (Scenario 3), we apply adaptive weights according to the precision of annotations.

The segmentation loss measures the similarity between the estimated probability map $\widehat{M}$ and the given masking image $M$. Popular options include the cross-entropy loss and dice coefficients. The cross-entropy loss measures the similarity between $\widehat{M}$ and $M$ at the pixel level, i.e.,

$$l_s(\hat{M}) = \sum_{1 \le i \le d_1} \sum_{1 \le j \le d_2} M_{i,j} \log \hat{M}_{i,j} + (1 - M_{i,j}) \log \left(1 - \hat{M}_{i,j}\right),$$

whereas the dice coefficient loss quantifies the overlap between the estimated map $\widehat{M}$ and the training mask $M$, and is given by

$$l_s(\hat{M}) = -\frac{\sum_{i,j} M_{i,j} \hat{M}_{i,j}}{\sum_{i,j} M_{i,j} + \sum_{i,j} \hat{M}_{i,j} - \sum_{i,j} M_{i,j} \hat{M}_{i,j}}.$$

The prediction loss varies depending on the type of learning problems. For classification problems, we adopt a cross-entropy loss that measures the KullbackLeibler divergence between the estimated probabilities and the observed values, i.e.,

$$l_p(\widehat{y}) = -\sum_{k=1}^{K} \left\{ y_k \log \widehat{y}_k + (1 - y_k) \log (1 - \widehat{y}_k) \right\},$$

where $\widehat{y} = [\widehat{y}_1, \ldots, \widehat{y}_K]^T$ represents the vector of predicted probabilities for the $K$ class labels, and $y_k$ is the categorical indicator, i.e., $y_k = 1$ if the object belongs to the $k$-th category. For regression problems, we may apply the squared loss, i.e., $l_p(\widehat{y}) = (\widehat{y} - y)^2$.

### 4.3.4 Implementation

We adopt a U-net (Ronneberger et al., 2015) structure as our segmentation module. We use a convolutional layer with kernel size $(3, 3)$ or $(3, 3, 3)$ for 2D or 3D input images, respectively, and apply zeros as padding to maintain fixed image sizes during convolutions. We add batch normalization (Ioffe and Szegedy, 2015) layers after each convolutional layer and activate the feature maps using rectifier liner units (Nair and Hinton, 2010). In the down-sampling phase of the network, we apply the maximum pooling layers with kernel size $(2, 2)$ or $(2, 2, 2)$, and use repetition up-sampling with the same kernel size to increase the resolution of feature maps in the up-sampling phase. The segmentation loss is the pixel wise cross-entropy function.

For the prediction module, we use a VGG structure (Simonyan and Zisserman, 2014) with batch normalized convolutional layers for 2D images, and a ResNet structure (He et al., 2016) for 3D images. For 2D images, we use the $\lambda = 10^{-4}$ as the weight parameter of the segmentation module and $\lambda = 10^{-6}$ for 3D images when rough annotations are provided.

We conduct the numerical experiments with training, validation, and test datasets, and use the stochastic gradient descent algorithm to train the networks (see (Ruder, 2016) for details about the algorithm). In particular, we use training sets to estimate model parameters and evaluate the model on the validation set at the end of each epoch. We initialize the learning rate of the algorithm with $10^{-4}$ and gradually decrease it if the validation loss does not decrease for ten consecutive epochs. The model with minimum validation loss is output as the final model and is used to evaluate the predictive accuracy on the test set.

In the numerical studies, we compare our model with the corresponding prediction network without the segmentation module, i.e., the model with the same structure as the prediction module in MCNN. We denote such a model as CNN to distinguish it from MCNN. For Scenarios 1-3, the MCNN models are denoted as MCNN 1-3 respectively, i.e., MCNN 1 for the case with precise annotations, MCNN 2 for imprecise annotations and MCNN 3 for mixed samples.

## 4.4 Synthetic simulation experiments

In this section, we conduct numerical studies with synthetic images, in which the segmentation ground truth are known. We generate the training mask according to the three scenarios discussed in Section 4.2, and apply the strategies in Section 4.3 to assign the loss weight.

### 4.4.1 Synthetic image regression

In this experiment, we simulate a set of symbolic images, each containing 3 ROIs: a circle, a square and a triangular region. The ROIs vary randomly by size and by location within a 32-by-32-pixel grid. The responses are generated according to the radius of the circles and the area of the squares. The triangular regions are not related to the responses. We add Gaussian noise with standard deviation 1 to each response and impose background noise to the covariate images according to a Gaussian random field. We generate $40,000$ training, $10,000$ validation, and $10,000$ test samples. The precise training masks in Scenario 1 are the images with pixel-level annotations of the corresponding ROIs. The rough masking images in Scenario 2 cover the predictive signals with larger irregular regions. For scenario 3, only 20% of the training samples have pixel-wise annotations, and the rest do not have any annotation information.

Some results are illustrated in Figure 4.2. The estimated masks clearly capture the predictive ROIs while ruling out the irrelevant triangular region when precise annotations are provided. With rough training masks, the predictive ROIs are still identifiable by the model and the background noise are reduced. In each of the three scenarios, the non-predictive triangular regions are effectively ruled out.

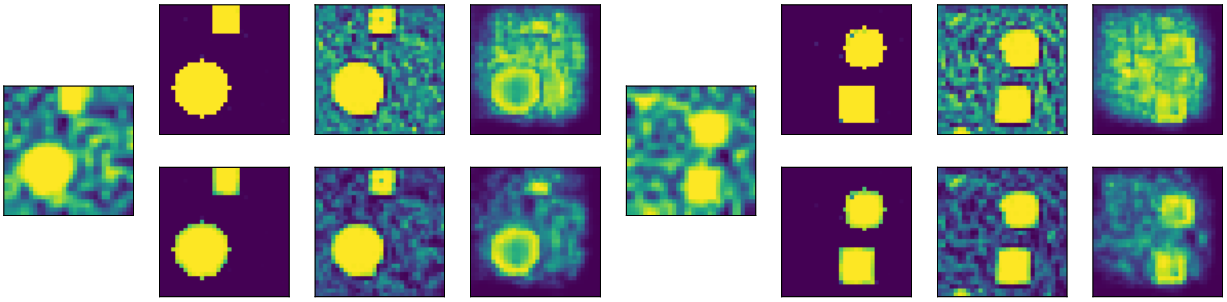


**Figure 4.2:** Estimation results for the synthetic image regression. In each of the two panels, the image on the left is the original input image. The three images on the top are the estimated masks for Scenario 1-3 respectively. The images on the bottom are the corresponding masked images.

### 4.4.2 Noisy MNIST

The MNIST (LeCun et al., 1998) dataset was constructed from a number of scanned documents collected by the National Institute of Standards and Technology. Each image is of size 28 by 28 pixels. There are $50,000$ training samples, $10,000$ validation samples, and $10,000$ test samples in the dataset. The original MNIST images have clean backgrounds and are relatively easy to classify with high accuracy, e.g., over 99.5% in (Wan et al., 2013). In order to evaluate the improvement of the proposed network structure, we conducted our experiment with noisy MNIST images that we created by adding random noise to the original MNIST images and randomly resizing the digits and shifting them within a 32-by-32-pixel grid. The precise training masks cover the digit regions in each image. The rough masks contain broader regions around the digits. For mixed samples, we assign 20% of the training samples with pixel-wise annotations, and the rest with no annotation information.

Figure 4.3 illustrates some of the estimation results, from which we can see that the estimated masking images can effectively block out background noise and accurately extract signals of digits from the noisy images. With the precise annotations, the extracted ROIs are more accurate, even when such annotations are available only for a small portion of samples.

## 4.5 Real applications

We also apply MCNN models in real image prediction problems, in which the ground truth of the masking images are unknown. We apply our prior knowledge about the data and create

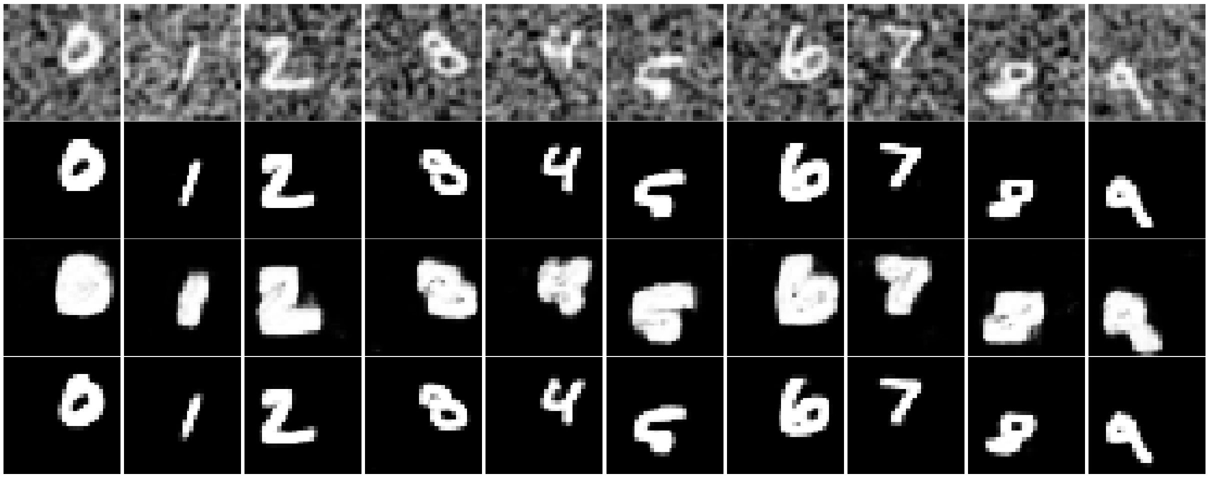


**Figure 4.3:** The estimation results for the noisy MNIST experiment. The top row is the plot of the input images with digits 0 to 9 from left to right. The following rows are the corresponding estimated masks from Scenario 1-3 respectively.

binary images that roughly cover the predictive regions as training masking images. Those masks are unique across different samples and are refined accordingly in the estimated results. Therefore, these experiments can be categorized to Scenario 2 in Section 4.2.

### 4.5.1 Street view house number (SVHN)

In this experiment, we classify the street view house number (SVHN) images (Netzer et al., 2011). This dataset consists of color images of house numbers collected by Google Street View. Each image is of size 32 × 32 pixels and may contain multiple digits. Our target is to classify the digit in the center of the image. The other digits must be ignored. There are $73,257$ images in the training set and $26,032$ images in the test set. We further split the training images into $50,000$ training samples and $23,257$ validation samples for model selection. We applied the training masking images that cover the middle part and ruled out the side regions of the input images. With such masking images, the model tends to focus on the center of the images so that any interference from the digits on the side can be reduced.

We illustrate the estimation results in Figure 4.4 and the training history in Figure 4.5. The estimated masks are able to highlight the target digits in the center while masking the digits on the sides. The test accuracy is improved from 90.73% to 95.13% compared to that achieved by the CNN model. Some of the incorrect classification of images is potentially due to incorrectly

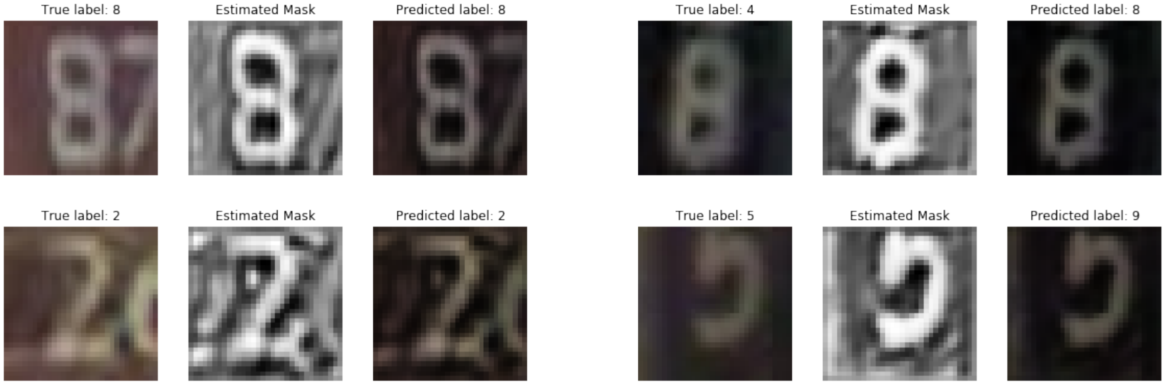


**Figure 4.4:** The estimation results for the SVHN experiment. The left group of images shows two correctly classified images, and the right group corresponds to the misclassified images. Each of the four panels consists of three images: the original noisy image, the estimated mask and the masked image.

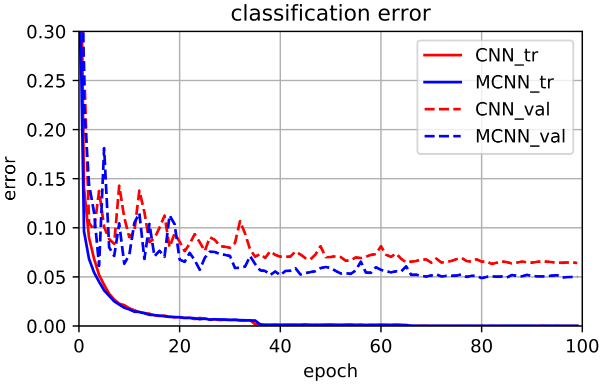


**Figure 4.5:** Training history of the CNN and MCNN models in the SVHN experiment. The red and blue lines represent the loss function value of the CNN and MCNN models, respectively. The solid and dashed lines correspond to the training and validation losses, respectively.

annotated labels (e.g., the first row in the right group of images) or incomplete digit regions (see the second row in the right group). The training history also indicates a stable validation error and improved classification accuracy.

### 4.5.2 ADNI MRI classification

In this experiment, we aim to classify the structural magnetic resonance imaging (MRI) data from the Alzheimer's Disease Neuroimaging Initiative (ADNI) study. The ADNI study was launched in 2003 as a large-scale, long-term project to collect MRI and positron emission tomography images, cerebrospinal fluid, and blood biomarkers, among other data. The goal of the ADNI study is to track the progression of Alzheimer's disease using these biomarkers and assess the brains structural and functional changes over different disease states. More information about this study can be found at the ADNI website (http://www.adni-info.org/).

We utilize the RAVENS-maps of the T1-weighted MRI images from different phases of the study, including ADNI1, ADNI2 and ADNI GO. The total number of participants in this study is 749, including participants with Alzheimer's disease, mild cognitive impairment, and healthy status. For each participant, multiple images are collected at different time points, and their disease status may vary as well. We use the disease status at the time of image acquisition as its corresponding class label. After dropping the images with no diagnostic results or low quality, we collect a total of $3,021$ images in our study. We generate the RAVENS-maps by following the pipeline in (Liu et al., 2018) and further down-sample the maps to the resolution of $64 \times 64 \times 64$ for the consideration of computational load. We randomly split the samples into training (80%), validation (10%), and test (10%) sets in the modeling procedure, and apply a mask that covers the whole brain region as the training image for each sample.

This experiment involves 3D images, and all the networks are built with 3D operations, including 3D convolution, up-sampling and pooling. In consideration of the model size, we use the ResNet structure (He et al., 2016) for the prediction phase.

Figure 4.6 illustrates some of the estimation results. We can see that the estimated masks tend to select most of the brain regions, while focusing on the frontal cortex, temporal gyrus, hippocampus, and fornix regions. These parts of the brain have been well studied in the literature and have been shown to relate to planning, logical thinking, and memory (Lue et al., 1999; Chan et al., 2001; Bordi et al., 2016; Lozano et al., 2016). With such masks, the classification accuracy is improved from 90.04% to 92.74% compared to that of the CNN model. From the plot of training losses in Figure 4.7, we can see that the loss of MCNN decreases more slowly at the beginning, but decreases more quickly after 60 epochs, and the validation results become more stable as well. This is mainly due to the larger model size compared to that of the CNN model. Once a stable estimation from the segmentation module is achieved, the classification results become better and more stable.

We summarize the prediction results of all the experiments in Table 4.1. Note that with the segmentation module added, the MCNN models tend to deliver better results in both regression and classification. Also note that, the more precise annotations, the better prediction we will have.

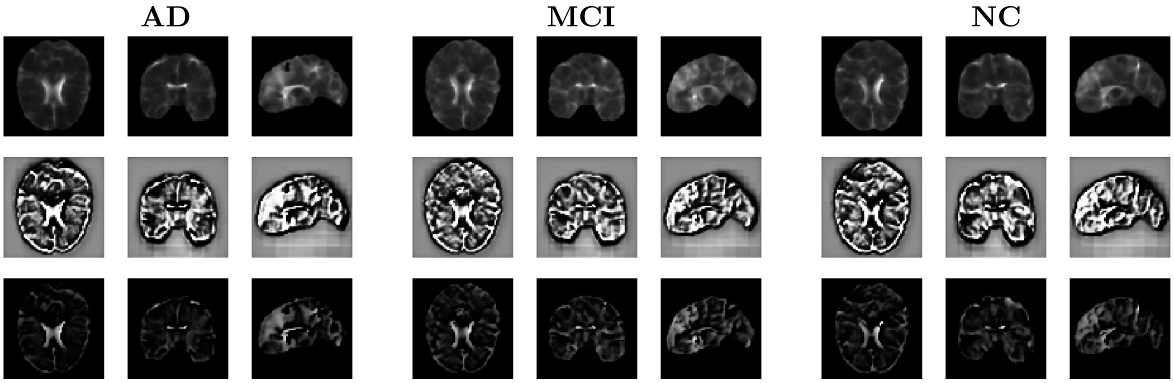


**Figure 4.6:** Estimation results for the ADNI experiment. The three groups of brain images respectively show typical samples of Alzheimers disease (AD), mild cognitive impairment (MCI) and healthy status (NC). From top to bottom, each column shows three images: the original image, the estimated mask, and the masked image.

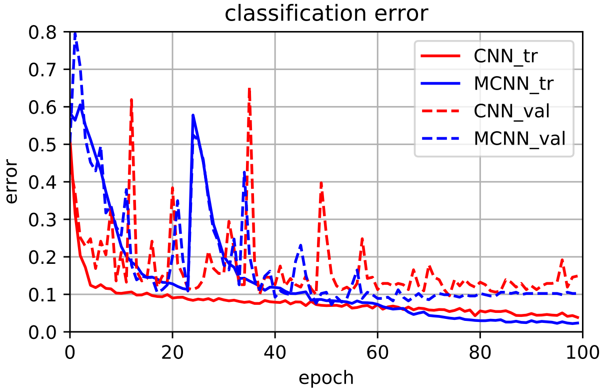


**Figure 4.7:** Training history of the CNN and MCNN models in the ADNI experiment. The red and blue lines represent the loss function values of the CNN and MCNN models, respectively. The solid and dashed lines correspond to the training and validation losses, respectively.

**Table 4.1:** Summary of results in the numerical experiments. The mean squared prediction errors are reported for the regression problem and the misclassification errors are reported for the classification problems.

| Method | SIM | MNIST | SVHN | ADNI |
|---|---|---|---|---|
| CNN | 2.27 | 5.30% | 9.27% | 9.60% |
| MCNN1 | **1.31** | **2.02%** | N/A | N/A |
| MCNN2 | 1.60 | 2.33% | **4.87%** | **7.26%** |
| MCNN3 | 1.51 | 2.28% | N/A | N/A |

## 4.6 Discussion

In this chapter, we propose an MCNN model that can simultaneously tackle segmentation and prediction problems. More importantly, the proposed model can generate masking images that are able to select the predictive ROIs in the input images and mask off the background noise. This can potentially enhance the target signals and improve the predictive accuracy.

The segmentation module in the MCNN model functions as the pre-whitening process for the input images. This is beneficial when the backgrounds are actual noise and not informative for the prediction. In some cases, the background signals and target objects are highly correlated and our method might not significantly improve the prediction.

We have focused on cases with non-informative backgrounds in the numerical experiments. The main purpose of these studies is to demonstrate the improved prediction achieved by MCNN due to the segmentation module. Thus, instead of comparing our proposed models results with benchmark results, we have mainly compared the MCNN models with the CNN model with the same structure as our prediction module. We do not have a specific requirement for the network structure. Any segmentation and prediction networks can be used to construct the MCNN models. However, including the segmentation module increases the size of the model. This can be a potential issue, but may be solved by parameter sharing, i.e., using the same structure and parameters in the down-sampling phase of both the segmentation and prediction modules. The competitive performance of MCNN both in terms of interpretability and accuracy suggests this is a promising area for future research.

# BIBLIOGRAPHY

Afonso, M. V., Bioucas-Dias, J. M., and Figueiredo, M. A. (2010). Fast image recovery using variable splitting and constrained optimization. *Image Processing, IEEE Transactions on*, 19(9):2345–2356.

Ashburner, J. and Friston, K. J. (2000). Voxel-based morphometrythe methods. *Neuroimage*, 11(6):805–821.

Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495.

Bassett, R. and Deride, J. (2016). Maximum a posteriori estimators as a limit of bayes estimators. *arXiv preprint arXiv:1611.05917*.

Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202.

Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 259–302.

Bordi, M., Berg, M. J., Mohan, P. S., Peterhoff, C. M., Alldred, M. J., Che, S., Ginsberg, S. D., and Nixon, R. A. (2016). Autophagy flux in ca1 neurons of alzheimer hippocampus: Increased induction overburdens failing lysosomes to propel neuritic dystrophy. *Autophagy*, 12(12):2467–2483.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122.

Bühlmann, P. and Van De Geer, S. (2011). *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media.

Busatto, G. F., Garrido, G. E., Almeida, O. P., Castro, C. C., Camargo, C. H., Cid, C. G., Buchpiguel, C. A., Furuie, S., and Bottino, C. M. (2003). A voxel-based morphometry study of temporal lobe gray matter reductions in alzheimers disease. *Neurobiology of aging*, 24(2):221–231.

Cai, T. T., Hall, P., et al. (2006). Prediction in functional linear regression. *The Annals of Statistics*, 34(5):2159–2179.

Carroll, M. K., Cecchi, G. A., Rish, I., Garg, R., and Rao, A. R. (2009). Prediction and interpretation of distributed neural activity with sparse models. *NeuroImage*, 44(1):112–122.

Casanova, R., Whitlow, C. T., Wagner, B., Williamson, J., Shumaker, S. A., Maldjian, J. A., and Espeland, M. A. (2011). High dimensional classification of structural mri alzheimers disease data based on large scale regularization. *Frontiers in neuroinformatics*, 5.

Chan, D., Fox, N. C., Scahill, R. I., Crum, W. R., Whitwell, J. L., Leschziner, G., Rossor, A. M., Stevens, J. M., Cipolotti, L., and Rossor, M. N. (2001). Patterns of temporal lobe atrophy in semantic dementia and alzheimer's disease. *Annals of neurology*, 49(4):433–442.

Chan, R. H., Nagy, J. G., and Plemmons, R. J. (1993). Fft-based preconditioners for toeplitz-block least squares problems. *SIAM journal on numerical analysis*, 30(6):1740–1768.

Chen, E., Chung, P.-C., Chen, C.-L., Tsai, H.-M., Chang, C.-I., et al. (1998). An automatic diagnostic system for ct liver image classification. *Biomedical Engineering, IEEE Transactions on*, 45(6):783–794.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2018). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848.

Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*.

Colliot, O., Chételat, G., Chupin, M., Desgranges, B., Magnin, B., Benali, H., Dubois, B., Garnero, L., Eustache, F., and Lehéricy, S. (2008). Discrimination between alzheimer disease, mild cognitive impairment, and normal aging by using automated segmentation of the hippocampus 1. *Radiology*, 248(1):194–201.

Cox, D. R. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 215–242.

Davatzikos, C., Genc, A., Xu, D., and Resnick, S. M. (2001). Voxel-based morphometry using the ravens maps: methods and validation using simulated longitudinal atrophy. *NeuroImage*, 14(6):1361–1369.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE.

Dubois, B., Feldman, H. H., Jacova, C., Hampel, H., Molinuevo, J. L., Blennow, K., DeKosky, S. T., Gauthier, S., Selkoe, D., Bateman, R., et al. (2014). Advancing research diagnostic criteria for alzheimer's disease: the iwg-2 criteria. *The Lancet Neurology*, 13(6):614–629.

Dukart, J., Mueller, K., Horstmann, A., Barthel, H., Möller, H. E., Villringer, A., Sabri, O., and Schroeter, M. L. (2011). Combined evaluation of fdg-pet and mri improves detection and differentiation of dementia. *PLoS One*, 6(3):e18111.

Ferraty, F. and Vieu, P. (2006). *Nonparametric functional data analysis: theory and practice*. Springer Science & Business Media.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188.

Fonov, V., Evans, A. C., Botteron, K., Almli, C. R., McKinstry, R. C., Collins, D. L., Group, B. D. C., et al. (2011). Unbiased average age-appropriate atlases for pediatric studies. *Neuroimage*, 54(1):313–327.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139.

Friedman, J., Hastie, T., Tibshirani, R., et al. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407.

Friston, K. J., Holmes, A. P., Worsley, K. J., Poline, J.-P., Frith, C. D., and Frackowiak, R. S. (1994). Statistical parametric maps in functional imaging: a general linear approach. *Human brain mapping*, 2(4):189–210.

Giedd, J. N., Blumenthal, J., Jeffries, N. O., Castellanos, F. X., Liu, H., Zijdenbos, A., Paus, T., Evans, A. C., and Rapoport, J. L. (1999). Brain development during childhood and adolescence: a longitudinal mri study. *Nature neuroscience*, 2(10):861–863.

Grosenick, L., Greer, S., and Knutson, B. (2008). Interpretable classifiers for fmri improve prediction of purchases. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 16(6):539–548.

Grosenick, L., Klingenberg, B., Greer, S., Taylor, J., and Knutson, B. (2009). Whole-brain sparse penalized discriminant analysis for predicting choice. *NeuroImage*, (47):S58.

Grosenick, L., Klingenberg, B., Katovich, K., Knutson, B., and Taylor, J. E. (2013). Interpretable whole-brain prediction analysis with graphnet. *NeuroImage*, 72:304–321.

Guo, R., Ahn, M., and Zhu, H. (2014). Spatially weighted principal component analysis for imaging classification. *Journal of Computational and Graphical Statistics*, (just-accepted):00–00.

Hadamard, J. (1902). Sur les problèmes aux dérivées partielles et leur signification physique. *Princeton university bulletin*, 13(49-52):28.

Hall, P., Horowitz, J. L., et al. (2007). Methodology and convergence rates for functional linear regression. *The Annals of Statistics*, 35(1):70–91.

Hall, P., Marron, J. S., and Neeman, A. (2005). Geometric representation of high dimension, low sample size data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(3):427–444.

Hastie, T., Tibshirani, R., Friedman, J., and Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85.

Haxby, J. V., Gobbini, M. I., Furey, M. L., Ishai, A., Schouten, J. L., and Pietrini, P. (2001). Distributed and overlapping representations of faces and objects in ventral temporal cortex. *Science*, 293(5539):2425–2430.

Haynes, J.-D. and Rees, G. (2006). Decoding mental states from brain activity in humans. *Nature Reviews Neuroscience*, 7(7):523–534.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hinrichs, C., Singh, V., Xu, G., Johnson, S. C., Initiative, A. D. N., et al. (2011). Predictive markers for ad in a multi-modality framework: an analysis of mci progression in the adni population. *Neuroimage*, 55(2):574–589.

Hirono, N., Mori, E., Ishii, K., Ikejiri, Y., Imamura, T., Shimomura, T., Hashimoto, M., Yamashita, H., and Sasaki, M. (1998). Frontal lobe hypometabolism and depression in alzheimer's disease. *Neurology*, 50(2):380–383.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Hoshikawa, T. (2013). Mixture regression for observational data, with application to functional regression models. *arXiv preprint arXiv:1307.0170*.

Huang, G., Liu, Z., Weinberger, K. Q., and van der Maaten, L. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, volume 1, page 3.

Hurn, M., Justel, A., and Robert, C. P. (2003). Estimating mixtures of regressions. *Journal of Computational and Graphical Statistics*, 12(1):55–79.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Kang, J., Reich, B. J., and Staicu, A.-M. (2016). Scalar-on-image regression via the soft-thresholded gaussian process. *arXiv preprint arXiv:1604.03192*.

Karnath, H.-O., Berger, M. F., Küker, W., and Rorden, C. (2004). The anatomy of spatial neglect based on voxelwise statistical analysis: a study of 140 patients. *Cerebral Cortex*, 14(10):1164–1172.

Khoo, V. S., Dearnaley, D. P., Finnigan, D. J., Padhani, A., Tanner, S. F., and Leach, M. O. (1997). Magnetic resonance imaging (mri): considerations and applications in radiotherapy treatment planning. *Radiotherapy and Oncology*, 42(1):1–15.

Killiany, R. J. and Albert, M. S. (1993). Temporal lobe regions on magnetic resonance imaging identify patients. *Arch Neurol*, 50:949–954.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Liu, L. Y.-F., Liu, Y., Zhu, H., Initiative, A. D. N., et al. (2018). Smac: Spatial multi-category angle-based classifier for high-dimensional neuroimaging data. *NeuroImage*.

Liu, M., Zhang, D., Shen, D., Initiative, A. D. N., et al. (2012). Ensemble sparse classification of alzheimer's disease. *NeuroImage*, 60(2):1106–1116.

Liu, Y. and Yan, B. (2017). Smooth image-on-scalar regression for brain mapping. *arXiv preprint arXiv:1703.05264*.

Liu, Y. and Yuan, M. (2011). Reinforced multicategory support vector machines. *Journal of Computational and Graphical Statistics*, 20(4):901–919.

Liu, Y., Zhang, H. H., and Wu, Y. (2011). Hard or soft classification? large-margin unified machines. *Journal of the American Statistical Association*, 106(493):166–177.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440.

Lopez, M., Ramirez, J., Gorriz, J., Salas-Gonzalez, D., Alvarez, I., Segovia, F., and Chaves, R. (2009). Neurological image classification for the alzheimer's disease diagnosis using kernel pca and support vector machines. In *Nuclear Science Symposium Conference Record (NSS/MIC), 2009 IEEE*, pages 2486–2489. IEEE.

Lozano, A. M., Fosdick, L., Chakravarty, M. M., Leoutsakos, J.-M., Munro, C., Oh, E., Drake, K. E., Lyman, C. H., Rosenberg, P. B., Anderson, W. S., et al. (2016). A phase ii study of fornix deep brain stimulation in mild alzheimers disease. *Journal of Alzheimer's Disease*, 54(2):777–787.

Lue, L.-F., Kuo, Y.-M., Roher, A. E., Brachova, L., Shen, Y., Sue, L., Beach, T., Kurth, J. H., Rydel, R. E., and Rogers, J. (1999). Soluble amyloid $\beta$ peptide concentration as a predictor of synaptic change in alzheimer's disease. *The American journal of pathology*, 155(3):853–862.

Marron, J., Todd, M. J., and Ahn, J. (2007). Distance-weighted discrimination. *Journal of the American Statistical Association*, 102(480):1267–1271.

Moeskops, P., Viergever, M. A., Mendrik, A. M., de Vries, L. S., Benders, M. J., and Išgum, I. (2016). Automatic segmentation of mr brain images with a convolutional neural network. *IEEE transactions on medical imaging*, 35(5):1252–1261.

Mota, J. F., Xavier, J. M., Aguiar, P. M., and Püschel, M. (2011). A proof of convergence for the alternating direction method of multipliers applied to polyhedral-constrained functions. *arXiv preprint arXiv:1112.2295*.

Mungas, D. (1991). Iii-office mental status testing: A practical guide. *Geriatrics*, 46(7).

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning.

Norman, K. A., Polyn, S. M., Detre, G. J., and Haxby, J. V. (2006). Beyond mind-reading: multi-voxel pattern analysis of fmri data. *Trends in cognitive sciences*, 10(9):424–430.

Ogawa, S., Lee, T.-M., Kay, A. R., and Tank, D. W. (1990). Brain magnetic resonance imaging with contrast dependent on blood oxygenation. *Proceedings of the National Academy of Sciences*, 87(24):9868–9872.

Pangman, V. C., Sloan, J., and Guse, L. (2000). An examination of psychometric properties of the mini-mental state examination and the standardized mini-mental state examination: implications for clinical practice. *Applied Nursing Research*, 13(4):209–213.

Parikh, N. and Boyd, S. (2013). Proximal algorithms. *Foundations and Trends in optimization*, 1(3):123–231.

Payan, A. and Montana, G. (2015). Predicting alzheimer's disease: a neuroimaging study with 3d convolutional neural networks. *arXiv preprint arXiv:1502.02506*.

Petersen, R. C., Thomas, R. G., Grundman, M., Bennett, D., Doody, R., Ferris, S., Galasko, D., Jin, S., Kaye, J., Levey, A., et al. (2005). Vitamin e and donepezil for the treatment of mild cognitive impairment. *New England Journal of Medicine*, 352(23):2379–2388.

Ramírez, J., Chaves, R., Górriz, J. M., Álvarez, I., López, M., Salas-Gonzalez, D., and Segovia, F. (2009). Functional brain image classification techniques for early alzheimer disease diagnosis. In *Bioinspired Applications in Artificial and Natural Computation*, pages 150–157. Springer.

Ramsay, J. O. and Silverman, B. W. (2005). *Functional data analysis*. Springer Series in Statistics. Springer, New York, second edition.

Rathore, S., Habes, M., Iftikhar, M. A., Shacklett, A., and Davatzikos, C. (2017). A review on neuroimaging-based classification studies and associated feature extraction methods for alzheimer's disease and its prodromal stages. *NeuroImage*.

Reiss, P. T., Huo, L., Zhao, Y., Kelly, C., and Ogden, R. T. (2015). Wavelet-domain regression and predictive inference in psychiatric neuroimaging. *The annals of applied statistics*, 9(2):1076.

Reiss, P. T. and Ogden, R. T. (2007). Functional principal component regression and functional partial least squares. *Journal of the American Statistical Association*, 102(479):984–996.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

Rosen, W. G., Mohs, R. C., and Davis, K. L. (1984). A new rating scale for alzheimer's disease. *The American journal of psychiatry*.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

Rudin, L. I., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268.

Rusinek, H., Endo, Y., De Santi, S., Frid, D., Tsui, W.-H., Segal, S., Convit, A., and de Leon, M. (2004). Atrophy rate in medial temporal lobe during progression of alzheimer disease. *Neurology*, 63(12):2354–2359.

Shen, D. and Zhu, H. (2015). Spatially weighted principal component regression for high-dimensional prediction. In *International Conference on Information Processing in Medical Imaging*, pages 758–769. Springer.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Smith, S. M., Jenkinson, M., Johansen-Berg, H., Rueckert, D., Nichols, T. E., Mackay, C. E., Watkins, K. E., Ciccarelli, O., Cader, M. Z., Matthews, P. M., et al. (2006). Tract-based spatial statistics: voxelwise analysis of multi-subject diffusion data. *Neuroimage*, 31(4):1487–1505.

Stein, C. M. (1981). Estimation of the mean of a multivariate normal distribution. *The annals of Statistics*, pages 1135–1151.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108.

Tibshirani, R. J. (2011). *The solution path of the generalized lasso.* Stanford University.

Tikhonov, A. N. (1943). On the stability of inverse problems. In *Dokl. Akad. Nauk SSSR*, volume 39, pages 195–198.

Toiviainen, P., Alluri, V., Brattico, E., Wallentin, M., and Vuust, P. (2014). Capturing the musical brain with lasso: Dynamic decoding of musical features from fmri data. *Neuroimage*, 88:170–180.

Van Gerven, M. A. and Heskes, T. (2012). A linear gaussian framework for decoding of perceived images. In *2012 Second International Workshop on Pattern Recognition in NeuroImaging*, pages 1–4. IEEE.

Vapnik, V. (2013). *The nature of statistical learning theory.* Springer science & business media.

Viele, K. and Tong, B. (2002). Modeling with mixtures of linear regressions. *Statistics and Computing*, 12(4):315–330.

Wahba, G. (1990). *Spline models for observational data.* SIAM.

Walker, S. H. and Duncan, D. B. (1967). Estimation of the probability of an event as a function of several independent variables. *Biometrika*, 54(1-2):167–179.

Wan, L., Zeiler, M., Zhang, S., Le Cun, Y., and Fergus, R. (2013). Regularization of neural networks using dropconnect. In *International Conference on Machine Learning*, pages 1058–1066.

Wang, H., Roa, A. C., Basavanhally, A. N., Gilmore, H. L., Shih, N., Feldman, M., Tomaszewski, J., Gonzalez, F., and Madabhushi, A. (2014). Mitosis detection in breast cancer pathology images by combining handcrafted and convolutional neural network features. *Journal of Medical Imaging*, 1(3):034003.

Wang, S., Huang, M., Wu, X., and Yao, W. (2016). Mixture of functional linear models and its application to co2-gdp functional data. *Computational Statistics & Data Analysis*, 97:1–15.

Watanabe, T., Kessler, D., Scott, C., Angstadt, M., and Sripada, C. (2014). Disease prediction based on functional connectomes using a scalable and spatially-informed support vector machine. *Neuroimage*, 96:183–202.

West, M. J., Coleman, P. D., Flood, D. G., and Troncoso, J. C. (1994). Differences in the pattern of hippocampal neuronal loss in normal ageing and alzheimer's disease. *The Lancet*, 344(8925):769–772.

Xu, Y., Jack, C., Obrien, P., Kokmen, E., Smith, G., Ivnik, R., Boeve, B., Tangalos, R., and Petersen, R. (2000). Usefulness of mri measures of entorhinal cortex versus hippocampus in ad. *Neurology*, 54(9):1760–1767.

Yamashita, O., Sato, M.-a., Yoshioka, T., Tong, F., and Kamitani, Y. (2008). Sparse estimation automatically selects voxels relevant for the decoding of fmri activity patterns. *NeuroImage*, 42(4):1414–1429.

Ye, G.-B. and Xie, X. (2011). Split bregman method for large scale fused lasso. *Computational Statistics & Data Analysis*, 55(4):1552–1569.

Yu, F. and Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*.

Yu, G., Liu, Y., Thung, K.-H., and Shen, D. (2014). Multi-task linear programming discriminant analysis for the identification of progressive mci individuals.

Yuan, M., Cai, T. T., et al. (2010). A reproducing kernel hilbert space approach to functional linear regression. *The Annals of Statistics*, 38(6):3412–3444.

Zhang, C. and Liu, Y. (2014). Multicategory angle-based large-margin classification. *Biometrika*, 101(3):625–640.

Zhang, Q. and Zhu, S. (2018). Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19:27–39.

Zhang, W., Li, R., Deng, H., Wang, L., Lin, W., Ji, S., and Shen, D. (2015). Deep convolutional neural networks for multi-modality isointense infant brain image segmentation. *NeuroImage*, 108:214–224.

Zhang, Y., Brady, M., and Smith, S. (2001). Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm. *IEEE transactions on medical imaging*, 20(1):45–57.

Zhu, J. and Hastie, T. (2005). Kernel logistic regression and the import vector machine. *Journal of Computational and Graphical Statistics*, 14(1).

Zhu, J., Zou, H., Rosset, S., and Hastie, T. (2009). Multi-class adaboost. *Statistics and its Interface*, 2(3):349–360.

Zhu, X., Suk, H.-I., and Shen, D. (2014). Sparse discriminative feature selection for multi-class alzheimers disease classification. pages 157–164.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.