# Structured Sparse Methods for Imaging Genetics

by

Tao Yang

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved December 2016 by the
Graduate Supervisory Committee:

Jieping Ye, Co-Chair
Guoliang Xue, Co-Chair
Jingrui He
Baoxin Li
Jing Li

ARIZONA STATE UNIVERSITY

May 2017

ABSTRACT

Imaging genetics is an emerging and promising technique that investigates how genetic variations affect brain development, structure, and function. By exploiting disorder-related neuroimaging phenotypes, this class of studies provides a novel direction to reveal and understand the complex genetic mechanisms. Oftentimes, imaging genetics studies are challenging due to the relatively small number of subjects but extremely high-dimensionality of both imaging data and genomic data. In this dissertation, I carry on my research on imaging genetics with particular focuses on two tasks—building predictive models between neuroimaging data and genomic data, and identifying disorder-related genetic risk factors through image-based biomarkers. To this end, I consider a suite of structured sparse methods—that can produce interpretable models and are robust to overfitting—for imaging genetics. With carefully-designed sparse-inducing regularizers, different biological priors are incorporated into learning models. More specifically, in the Allen brain image–gene expression study, I adopt an advanced sparse coding approach for image feature extraction and employ a multi-task learning approach for multi-class annotation. Moreover, I propose a label structured-based two-stage learning framework, which utilizes the hierarchical structure among labels, for multi-label annotation. In the Alzheimer's disease neuroimaging initiative (ADNI) imaging genetics study, I employ Lasso together with EDPP (enhanced dual polytope projections) screening rules to fast identify Alzheimer's disease risk SNPs. I also adopt the tree-structured group Lasso with MLFre (multi-layer feature reduction) screening rules to incorporate linkage disequilibrium information into modeling. Moreover, I propose a novel absolute fused Lasso model for ADNI imaging genetics. This method utilizes SNP spatial structure and is robust to the choice of reference alleles of genotype coding. In addition, I propose a two-level structured sparse model that incorporates gene-level networks through a graph penalty into SNP-level model

construction. Lastly, I explore a convolutional neural network approach for accurate predicting Alzheimer's disease related imaging phenotypes. Experimental results on real-world imaging genetics applications demonstrate the efficiency and effectiveness of the proposed structured sparse methods.

*To my parents*

*for their love, endless support*

*and encouragement.*

# ACKNOWLEDGMENTS

First and foremost, I would like to gratefully and sincerely acknowledge my Ph.D. advisor, Dr. Jieping Ye, for his excellent guidance, continuous support, and kindly encouragement during my dissertation research. This dissertation would have never been possible without his help.

I would also like to express my sincere gratitude to my dissertation committee co-chair, Dr. Guoliang Xue. And many thanks to my dissertation committee members: Dr. Jingrui He, Dr. Baoxin Li, and Dr. Jing Li. Thanks for serving on my dissertation committee and for their valuable interactions and feedback.

It has been my great fortune and pleasure to collaborate with Dr. Paul Thompson, Dr. Li Liu, Dr. Liang Zhan, Dr. Yalin Wang, Dr. Vaibhav Narayan, and Dr. Gayle Wittenberg during my dissertation research. Thanks for their guidance, support, and insightful comments.

Last but not least, my colleagues and friends in Dr. Jieping Ye's research group inspired me a lot through discussions, seminars, and project collaborations. I would like to thank the following people for their valuable interactions: Dr. Jun Liu, Dr. Shuiwang Ji, Dr. Lei Yuan, Dr. Rita Chattopadhyay, Dr. Jiayu Zhou, Dr. Sen Yang, Dr. Shuo Xiang, Dr. Qian Sun, Cheng Pan, Rashmi Dubey, Xinlin Zhao, Yashu Liu, Zhi Nie, Qingyang Li, and Shuang Qiu. Many thanks to the postdocs in our group: Dr. Zheng Wang, Dr. Jie Wang, Dr. Chao Zhang, Dr. Binbin Lin, Dr. Pinghua Gong, Dr. Kefei Liu, Dr. Ming Lin and Dr. Yan Li.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Chapter 1

# INTRODUCTION AND BACKGROUND

## 1.1 Imaging Genetics

In the past decade, imaging genetics has attracted increasing attention. It has been widely recognized by molecular geneticists that some common genetic variants in *single nucleotide polymorphisms* (SNPs) could lead to common disorders [Cirulli and Goldstein (2010)]. Imaging genetics studies disorder-related genetic variation by taking advantage of neuroimaging phenotypes, as imaging phenotypes are closer to the biology of genetic function than disease or cognitive phenotypes [Meyer-Lindenberg (2012)] . With the involvement of molecular genetics and disorder-related neuroimaging phenotypes, imaging genetics provides a unique opportunity to reveal and understand the impact of genetic variation, *i.e.*, how individual differences in terms of SNPs affect brain development, structure, and function [Hariri *et al.* (2006); Thompson *et al.* (2013)].

Previous studies demonstrate the great promise of imaging genetics. For instance, on Chromosome 19, the $\epsilon 4$ allele of gene *Apolipoprotein E* (*a.k.a.*, ApoE4) is one of the well-known genetic risk factors for Alzheimer's disease (AD). From the neuroimaging perspective, the degeneration of brain tissue of ApoE4 carriers is faster as they age; young adults ApoE4 carriers often exhibit thinner cortical gray matter than noncarriers [Shaw *et al.* (2007)]. In the meantime, as has been verified in a series of genome-wide association (GWA) studies of AD, ApoE4 is strongly associated with the volumes of key brain regions, such as the hippocampus and entorhinal cortex [Potkin *et al.* (2009); Stein *et al.* (2012); Yang *et al.* (2015b)]. More recently,

worldwide consortium efforts, such as ENIGMA (Enhancing Neuroimaging Genetics through Meta-Analysis, [Stein *et al.* (2012)] and CHARGE (Cohorts for Heart and Aging Research in Genomic Epidemiology, [Bis *et al.* (2012); Psaty *et al.* (2009)]), enable us to investigate robust common neuroimaging genetic associations [Medland *et al.* (2014)].

Oftentimes in practice, imaging genetics studies are challenging due to the relatively small number of subjects but extremely high dimensionality of both neuroimaging data and genomic data. For example, neuroimaging data sets are typical of very high-resolution, in which an image file may contain hundreds of thousands of voxels (or imaging phenotypes). As a consequence, it could be extremely hard to identify or extract disorder-related phenotypes from the raw image data. In the meantime, advances in modern sequencing techniques lead to the huge scale of genome sequencing data. Typically, an SNPs data set may contain millions of loci (or say *base pairs*, *i.e.*, positions on the genome). The two facts mentioned above significantly limit the practical usage of traditional learning methods, as they are not effective in the high-dimensional scenario (*e.g.*, prone to overfitting).

There have been several practical attempts on imaging genetics. Indeed, we can categorize existing methodological approaches of imaging genetics into three classes [Thompson *et al.* (2013)]:

- *Univariate-imaging univariate-genetic association* analysis. This class of approaches performs a univariate statistical test on each SNP-voxel pair individually. It has been widely used in previous GWA studies. However, it fails to reveal scenarios such as the joint effects of multiple SNPs or SNP-SNP interactions, which occur commonly during gene expression [Singh *et al.* (2011); Dinu *et al.* (2012); Cornelis *et al.* (2009); Yang *et al.* (2012a)]. For genetics, the aggregate effects are usually more significant than individual effects. Moreover, it

is worth indicating that this kind of approaches is computationally inefficient.

- *Univariate-imaging multivariate-genetic association* analysis. Based on a pre-selected candidate imaging phenotype, a typical multivariate approach utilizes sparse models, *e.g.*, Lasso (least absolute shrinkage and selection operator, [Tibshirani (1996); Yang *et al.* (2015b)]), to perform simultaneous model fitting and model selection (*i.e.*, identify causal SNPs). There are also attempts to incorporate different biological prior knowledge during model constructions. For example, Wang *et al.* (2012) employ group Lasso [Yuan and Lin (2006)] to locate groups of candidate SNPs, where SNP groups are pre-defined by linkage disequilibrium (LD) information.

- *Joint multivariate association* analysis. This class of approaches investigates the relationships between two sets of variables; for example, canonical correlation analysis (CCA) and partial least squares (PLS) regression. However, as a side note, a clear drawback is that the detected genetic variants and imaging phenotypes may not be immediately related to a disorder [Batmanghelich *et al.* (2013)].

In this dissertation, I concentrate on the second class of methodologies, *i.e.*, the univariate-imaging multivariate-genetic association approaches, for imaging genetics. More specifically, I carry on my research with two particular focuses in imaging genetics: 1) building predictive models between genomic data and neuroimaging phenotypes, and 2) identifying disorder-related genetic risk factors through image-based biomarkers. Research work presented in this dissertation are primarily based on my previous work: Wang *et al.* (2016a); Yang *et al.* (2015c,b, 2016); Li *et al.* (2016b,a).

In the next section, I brief introduce a suite of structured sparse methods for addressing imaging genetics problems.

## 1.2 Structured Sparse Methods

Most traditional statistical learning methods are intended for the low-dimensional scenario [James *et al.* (2013)], where the number of subjects $n$ is usually much larger than the number of features $p$. However, in imaging genetics studies, we usually have $p \gg n$, where $p$ refers to the feature dimension of the genomic data. As a consequence, traditional learning method cannot produce desired predictive performance, as they are prone to overfitting in the high-dimension scenario.

The high-dimensional data involved in imaging genetics confront researchers and scientists with an urgent request for novel methods that can effectively reveal the predictive patterns under such a circumstance. A useful observation from many real-world applications is that data set with complex structures often has sparse underlying representations. More specifically, although the data may have millions of features, it may be well interpreted by a few most relevant explanatory features. For example, the neural representation of natural scenes in the visual cortex is sparse, as only a small number of neurons are active at a given instant [Vinje and Gallant (2000)]; images have very sparse representations with respect to an over-complete dictionary because they lie on or close to low-dimensional subspaces or submanifolds [Wright *et al.* (2010)]; although humans have millions of SNPs, only a small number of them are relevant to certain diseases such as leukemia, Alzheimer's disease [Golub *et al.* (1999); Guyon *et al.* (2002); Mu and Gage (2011)]. In addition, sparsity has been shown to be an effective approach to alleviate overfitting, from which most traditional statistical approaches suffer. Therefore, finding sparse representations is particularly significant in revealing underlying mechanisms of many complex systems.

In the past decade, as an emerging and promising technique, sparse methods has attracted increasing research interests in image genetics. As a class of regularized

model learning approach, sparse models are typically robust to overfitting. Meanwhile, sparse approaches can enhance the model interpretability, as only a small subset of features, which can best explain the outcome, will be identified. In addition, it is worth mentioning that, by utilizing carefully-designed sparse-inducing regularizers, we can incorporate different biological prior knowledge into the models. This is beneficial, since complex feature structures, such as LD information and SNP spatial structure, can be introduced during model construction.

In Chapter 2, I will briefly review several existing structured sparse methods and introduce some related optimization methods. For the two real-world imaging genetics applications, I consider a suite of structured sparse methods. More specifically, in the Allen brain image—gene expression study, I propose to use sparse coding for image feature extraction and employ group Lasso for multi-class annotation. I also introduce a two-stage multi-label learning framework based on label hierarchical structure. In the ADNI imaging genetics study, I propose to incorporate linkage disequilibrium information through tree-structured group Lasso, and incorporate SNP spatial information through a novel absolute fused Lasso. In the sequel, I propose a two-level structured sparse model that acts as a bridge to connect genes and SNPs as well as utilizing gene networks. In addition, I present a convolutional neural network with dropout for accurate predicting Alzheimer's disease related imaging phenotypes. In the rest parts of this chapter, I briefly introduce the background of the aforementioned two research works.

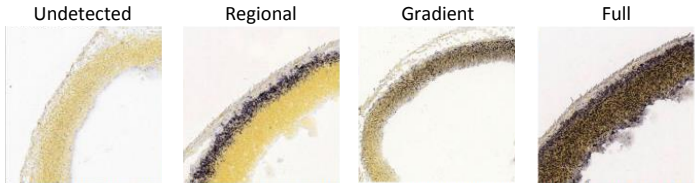
## 1.3 Background of Allen Brain Imaging – Gene Expression Study

Brain tumor is a fatal central nervous system disease. It is also the second cause of cancer in children [World Health Organization (2014)]. Previous studies indicate that preventing and detecting brain tumors at early stages are effective methods to

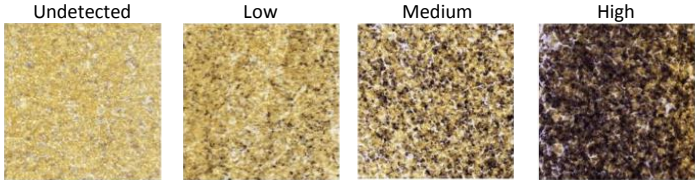
reduce brain damage; these studies also show the potential benefit of utilizing the genetic determinants [Reilly (2009)]. Accurate descriptions of the locations of where the relative genes are active and how these genes express are critical for understanding the pathogenesis of brain tumor and for early detection. In this study, I investigate the associations between gene expression patterns and brain images on the Allen developing mouse brain atlas (ADMBA) [Allen Institute for Brain Science (2013)]. ADMBA is an online public repository that contains extensive gene expression data and neuroanatomical data over different mouse brain developmental stages.

ADMBA provides extensive experimental resources of the brain. For imaging data, the atlas stores about 435,000 high-resolution spatiotemporal *in-situ Hybridization* (ISH) brain images from embryonic through postnatal stages of mouse development. Those images cover approximately 2,100 genes at each stage. Meanwhile, a brain ontology has been designed to hierarchically organize brain structure. To categorize the gene expression status at certain brain regions revealed by in situ Hybridization images, three kinds of measurements—*i.e.*, pattern, density, and intensity—are employed at the reference atlas for ADMBA (R-ADMBA). These measurements were scored for each brain region according to a set of standard schemas; examples are shown in Figure 1.1.

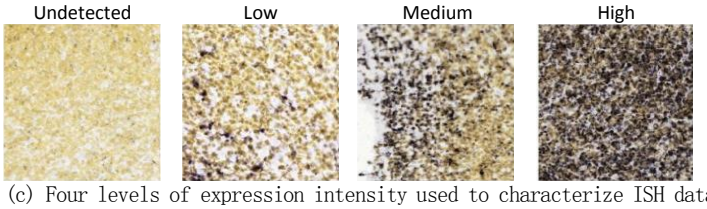
It is worth mentioning that such annotation tasks are very costly, since the entire atlas contains more than four million ISH images, and there are about one thousand brain regions that need to be annotated in the designed brain ontology. To precisely assign gene expression status to specific brain regions, current reference atlas uses expert-guided manual annotation, which was performed by Dr. Martinez's team at Spain [Allen Institute for Brain Science (2013); Thompson *et al.* (2014)]. However, it is labor-intensive since it requires expertise in neuroscience and image analysis. In other words, it does not scale with the continuously expanding collection of images.



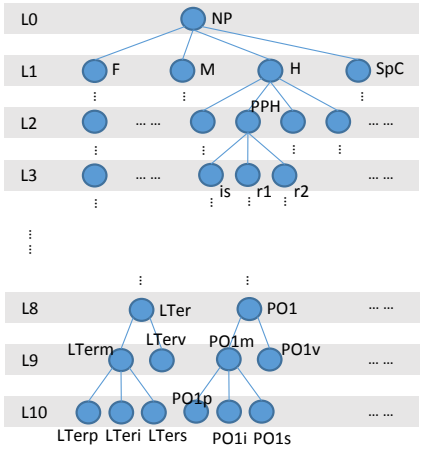

(a) Four types of expression patterns used to characterize ISH data




(b) Four levels of expression density used to characterize ISH data




(c) Four levels of expression intensity used to characterize ISH data




(d) Part of Brain Ontology

* Detailed ontology of each node can be found in: http://developingmouse.brain-map.org/docs/Legend_2010_03.pdf

**Figure 1.1:** (a-c) Sample Schemas of Three Annotation Measurements (Pattern / Density / Intensity) associated with Four Expression Levels. (d) Part of Brain Ontology.

Therefore, developing an effective and efficient automated gene expression pattern annotation method is of practical significance.

In this dissertation, I introduce a series of brain image annotation studies associated with ADMBA. The major target of those studies is to develop a computational framework to perform automated gene expression patterns annotation over the entire brain ontology based on a suite of high-resolution spatiotemporal in-situ Hybridization brain images. Specifically, there are three major challenges in these studies:

- High-resolution spatiotemporal images: around 435,000 unaligned ISH images; up to 12 million pixels per image; over 2,000 different genes;
- Multi-class annotation: 4 different expression levels;
- Multi-label annotation: 1,025 topological subdivisions (regions) over brain.

For the first challenge, traditional approaches [Zeng and Ji (2014)] utilize the scale-invariant feature transform (SIFT) [Lowe (1999)] algorithm and the bag-of-words (BoW) [Csurka *et al.* (2004)] model to extract patch-level image characteristics and learn high-level image representations, respectively. It is worth mentioning that BoW is not efficient to learn a large number of keywords or deal with large scale data atlas. However, due to the huge size of image atlas as well as the complex brain ontology, a large keyword size is desired in this study. Besides the image representation problem, many other difficulties are also inherent in the annotation tasks. First of all, for a set of ISH images and a specific measurement, current reference atlas uses up to four categories [see Figure 1.1, (a-c)] to give an accurate description of the gene expression status. That is, such an annotation task is indeed a multi-class classification problem. Secondly, annotating gene expression pattern over the entire brain ontology is essentially a multi-label classification problem, since there are over one thousand brain regions to be annotated. However, for multi-label annotation, if we do not take label dependency into consideration—*i.e.*, simply treat each label separately, it may result in suboptimal prediction performance [Silla Jr and Freitas (2011); Tsoumakas *et al.* (2010); Bi and Kwok (2011)].

In this dissertation, I adopt structured spares methods in three major sub-tasks. Specifically, in Section 3.1, I adopt an augmented sparse coding method to construct high-level image features during image feature processing; in Section 3.2, I utilize the $\ell_{2,1}$-norm regularized logistic regression model for the multi-class annotation problem; in Section 3.3, I propose a novel two-stage learning framework based on label hierarchy structure to improve the annotation accuracy over the entire brain ontology (i.e., the multi-label problem).

## 1.4 Background of ADNI Imaging Genetics Study

Alzheimer's disease (AD) is the most common form of dementia; it affects more than five million Americans and is the sixth-leading cause of death in the United States [Hebert *et al.* (2013)]. AD is an irreversible, progressive brain disorder typically beginning with mild memory loss; later it can seriously impair an individual's ability to carry out daily activities. It has been widely recognized and emphasized that early detection of AD is beneficial. Recently, neuroimaging techniques—such as magnetic resonance imaging (MRI), computed tomography (CT) and positron emission tomography (PET)—have shown great promise for evaluating AD and tracking its progression [Weiner *et al.* (2005)].

Factors that influence AD progression are not yet fully understood, but common genetic variants are among the major risk factors [Huang and Mucke (2012)]. Novel sequencing techniques have greatly advanced genome-wide association studies (GWAS) and whole genome sequencing (WGS) studies. More recently, entire genomes can be combined with brain imaging and clinical data to facilitate the investigation of mechanisms of AD.

Besides the well-known APOE genotype, recent studies [Bettens *et al.* (2013)] of the Alzheimer's disease neuroimaging initiative (ADNI) GWAS data have related known AD risk genes to differences in rates of brain atrophy and biomarkers of AD in the cerebrospinal fluid. More recently, full genetic sequences have been collected for over one thousand ADNI participants. The ENIGMA Consortium recently discovered six common genetic variants associated with subcortical brain volumes in a worldwide screen of over 30,000 brain MRI scans [Stein *et al.* (2012)]. Another studythe international genomics of Alzheimer's project (I-GAP) study [Lambert *et al.* (2013)], with over 74,000 participantsidentified genetic risk factors with statistical methods,

is the largest study of AD to date. However, these studies still have limitations. First of all, GWAS studies focus on a set of selected common genetic variants rather than the entire sequence of the genome. Loci showing strongest associations with the disease, or a brain measure, are not often the causal SNPs, as the causal loci have typically not been sequenced directly. Secondly, studies such as I-GAP are based on simple statistical models that only test associations of each SNP, one at a time, with AD-related phenotypes. In other words, these methods typically ignore potential inter-locus interactions.

In this dissertation, I focus on a series of imaging genetic studies that aim to investigate the associations between ADNI T1 MRI data and WGS data, i.e., how genetic variants, in terms of SNPs, affect the progression of AD. As mentioned in Section 1.1, my research is carry on with two major focuses: 1) building predictive models between genomic data and neuroimaging phenotypes, and 2) identifying disorder-related genetic risk factors through image-based biomarkers. To this end, I adopt Lasso as a basic multivariate method, together with a suite of structured sparse methods, which are capable of incorporating different biological prior knowledge, to reveal AD-related SNPs. Specifically, in Section 4.1, I present the basic Lasso model and its screening approach for the ADNI imaging genetics study; in Section 4.2, I employ the tree-structured group Lasso to incorporate linkage disequilibrium information; in Section 4.3, I propose to incorporate SNP spatial information through a novel absolute fused Lasso; in Section 4.4, I propose a two-level structured sparse model that acts as a bridge to connect gene-level descriptors and low-level SNPs, as well as utilizing gene networks information; in Section 4.5, I present a convolutional neural network with dropout layers for accurate predicting Alzheimer's disease related imaging phenotypes.

Chapter 2

# STRUCTURED SPARSE METHODS

In this chapter, I brief review some existing approaches for studying imaging genetics. Specifically, I first introduce two simple sparse models based on $\ell_1$-norm. Next, I discuss several structure-based sparse models, including group Lasso, tree-structured group Lasso and the fused Lasso, for imaging genetics applications. Then, I review some optimization methods for solving the related convex and non-convex optimization problems. In the sequel, I introduce the idea of screening for boosting the computational efficiency. In the last part, I discuss some approaches of utilizing label structures in multi-label learning.

## 2.1 Basic Sparse Models

In the first section, I begin with some fundamental ideas of linear models. Given a training data set $\mathbf{A} \in \mathbb{R}^{n \times p}$ with $n$ observations and $p$ features, and $\mathbf{A} = [\mathbf{a}^1, \ldots, \mathbf{a}^n]^T = [\mathbf{a}_1, \ldots, \mathbf{a}_p]$. By convention, each row $\mathbf{a}^i \in \mathbb{R}^p$, $i = 1, \ldots, n$, represents a subject and each column $\mathbf{a}_j \in \mathbb{R}^n$, $j = 1, \ldots, p$, represents a feature vector. Let $\mathbf{y} \in \mathbb{R}^n$ denote a corresponding response vector of $\mathbf{A}$. Suppose $\mathbf{A}$ is centered and scaled, then a basic linear model $h : \mathbb{R}^p \to \mathbb{R}$ can be considered as follows:

$$h(\mathbf{A}) = \mathbf{x}^T \mathbf{A}, \tag{2.1}$$

where $\mathbf{x} \in \mathbb{R}^p$ is the coefficient vector that needs to be estimated.

Many traditional regression and classification methods like least squares and logistic regression are developed for the low-dimensional scenario, in which the number

of observations $n$ is (much) larger than the number of features $p$ [James *et al.* (2013)]. However, in many real-world applications, we frequently confront with some data sets that exhibit extremely high-dimensionality, where the number of features $p$ is much larger than the number of observations $n$. When $p >> n$, traditional methods may be not suitable due to the poor prediction performance (*a.k.a.* overfitting) or poor interpretability.

In the high-dimensional scenario, regularized approaches have been shown to be promising to alleviate overfitting as well as improve the model interpretability. By incorporating a sparse-inducing regularizer, the class of sparse learning models that estimates the coefficient $\mathbf{x}$ can be defined as follows:

$$\min_{\mathbf{x}} f(\mathbf{x}) = \ell(\mathbf{x}) + \lambda\Omega(\mathbf{x}), \tag{2.2}$$

where $\ell(\cdot)$ is a proper convex empirical loss function that measures the fitness of the model on the training data, $\Omega(\cdot)$ is a regularizer that penalizes the complexity of the model, and $\lambda \geq 0$ is a regularization parameter that controls the trade-off between the loss $\ell(\cdot)$ and the penalty $\Omega(\cdot)$. In addition, in many sparse approaches, the sparse-inducing penalty $\Omega(\cdot)$ is typically non-smooth and non-differentiable.

In this dissertation research, I consider two simple but extensively used sparse models: Lasso, which is for regression tasks, and sparse logistic regression, which is for classification tasks.

Lasso (least absolute shrinkage and selection operator) is a widely used regression technique to find sparse representations of a given signal with respect to a set of basis vectors [Tibshirani (1996)]. Standard Lasso employs least squares loss and $\Omega(\cdot) = \|\cdot\|_1$ as its regularizer, *i.e.*,

$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|^2 + \lambda\|\mathbf{x}\|_1. \tag{2.3}$$

Due to the properties of $\ell_1$-norm, many components in the coefficient vector will be

zeros when the value of $\lambda$ is large. The features corresponding to these non-zero components can be considered to be important to explain the outcome. Compared to Lasso, sparse logistic regression also adopts the $\ell_1$-norm regularizer but utilizes the logistic loss, which is designed specifically for classification tasks. It takes the form:

$$\min_{\mathbf{x}} \sum_{i=1}^{n} \log\left(\frac{1}{1+e^{-y_i(\mathbf{x}^T\mathbf{a}^i)}}\right) + \lambda\|\mathbf{x}\|_1. \tag{2.4}$$

Sparse logistic regression has attracted much attention in the past few years and the interest is growing due to the increasing number of high-dimensional data sets [Sun *et al.* (2009); Wu *et al.* (2009); Zhu and Hastie (2004)]. It is worth mentioning that both Lasso and sparse logistic regression can perform simultaneously model fitting (regression or classification) and variable selection, which has achieved great success in many real-world applications [Chen *et al.* (2001); Candès *et al.* (2006); Zhao and Yu (2006); Bruckstein *et al.* (2009); Wright *et al.* (2010)].

## 2.2 Structured Sparse Models

A major drawback of the aforementioned two sparse methods—Lasso and sparse logistic regression—is that they do not take feature structures into consideration. In other words, sparse representations obtained by Lasso or sparse logistic regression remain the same if we shuffle the order of features. However, in many real-world applications, this is undesirable, as the features often exhibit some certain intrinsic structures, *e.g.*, disjoint or overlapping feature groups, spatial and/or temporal smoothness, tree structure and graph structure. In this section, I introduce several structured sparse models, which are capable of incorporating different prior knowledge of features through carefully-designed sparse-inducing regularizers.

### 2.2.1 Group Lasso and Sparse Group Lasso

One scenario that occurs commonly in real-world applications is that features may form groups or clusters. For example, features with discrete values are usually transformed into groups of dummy variables; in a previous AD study, people divide the voxels of PET images into a set of non-overlapping groups according to the brain regions. Hence, in order to select groups of features, Yuan and Lin (2006) proposed the non-overlapping group Lasso (GL). Assume that the features are partitioned into $k$ disjoint groups $\{G_1, \ldots, G_k\}$, where $G_i$ contains the indices of features belonging to the $i^{th}$ group. The group Lasso regularizer takes the form of:

$$\Omega_{\text{gL}}(\mathbf{x}) = \sum_{i=1}^{k} w_i \|\mathbf{x}_{G_i}\|_q, \tag{2.5}$$

where $w_i$ is the weight for the $i^{th}$ group and $\|\cdot\|_q$ with $q > 1$ is the $\ell_q$-norm (the value of $q$ is usually set to be 2 or $\infty$) [Wang *et al.* (2013)]. Group Lasso has been widely used in applications when group structure is available, *e.g.*, regression [Kowalski (2009); Negahban and Wainwright (2008)], classification [Meier *et al.* (2008)], joint covariate selection for group selection [Obozinski *et al.* (2007)], and multi-task learning [Argyriou *et al.* (2008); Liu *et al.* (2009a); Quattoni *et al.* (2009)].

For some applications, it is desirable to determine features within each group that exhibit the strongest effects. To this end, the sparse group Lasso (SGL) [Friedman *et al.* (2010); Simon *et al.* (2013)] is preferred. SGL combines the group Lasso penalty and the Lasso penalty to identify important feature groups and features simultaneously. Specifically, SGL penalty can be written as:

$$\Omega_{\text{sgL}}(\mathbf{x}) = \alpha\|\mathbf{x}\|_1 + (1-\alpha)\sum_{i=1}^{k} w_i \|\mathbf{x}_{G_i}\|_q, \tag{2.6}$$

where $\alpha \in [0, 1]$ balances the sparsity in the feature-level and the sparsity in the group-level. In recent years, SGL has achieved great success in many real-world applications,

including machine learning [Vidyasagar (2014); Yogatama and Smith (2014)], signal processing [Sprechmann *et al.* (2011)], bioinformatics [Peng *et al.* (2010)], etc.

### *2.2.2 Overlapping Group Lasso and Tree Lasso*

Group Lasso assumes that the feature groups are disjoint. However, in some applications, some features may be shared across different groups. For example, in gene ontology studies, a gene may be involved in different biological pathways, which means it is shared across different groups [Ashburner *et al.* (2000); Harris *et al.* (2004); Subramanian *et al.* (2005)]. To this end, the overlapping group Lasso (OGL) penalty is desired. OGL penalty is similar to (2.5), but $G_i$ may overlap with $G_j$ when $i \neq j$ [Zhao *et al.* (2009)].

A particularly attracting special case of the overlapping group Lasso is the so-called tree-structured group Lasso [Kim and Xing (2010); Zhao *et al.* (2009)]. In certain real-world applications, the data may exhibit hierarchical tree-structured sparse patterns among features. For example, based on the spatial locality [Liu and Ye (2010)], we can represent an image by a tree where a leaf node corresponds to a single feature (pixel) and an internal node corresponds to a group of features (pixels). When the tree structure is available, we can formulate tree-structured group Lasso (TGL) as follows:

$$\Omega_{\text{tgL}}(\mathbf{x}) = \sum_{i,j} w_j^i \|\mathbf{x}_{G_j^i}\|_q, \tag{2.7}$$

where $G_j^i$ is the group of features corresponding to the $j^{th}$ node at depth $i$ and $w_j^i$ is the positive weight for $G_j^i$. We note that every node in the tree is a superset of its descendant nodes. As a consequence, if the features in a node are excluded from the sparse representation, so are the features in all its descendant nodes.

### 2.2.3 *Fused Lasso and Graph Lasso*

Another common scenario that happens in many real-world studies is that the data sets we investigated are of some natural order, *i.e.*, the features may come with spatial and/or temporal smoothness. For example, in studies of arrayCGH [Tibshirani *et al.* (2005); Tibshirani and Wang (2008)], the features—DNA copy numbers along the genome—exhibit a natural spatial order. To this end, the fused Lasso (FL) penalty is proposed to encode the structure of smoothness by penalizing the differences between adjacent coefficients, *i.e.*,

$$\Omega_{\text{fL}}(\mathbf{x}) = \alpha\|\mathbf{x}\|_1 + (1-\alpha)\sum_{i=1}^{p-1}|x_i - x_{i-1}|, \tag{2.8}$$

where $\alpha \in [0,1]$. It is clear that the fused Lasso penalty would lead to solutions in which adjacent components are close or identical to each other.

In certain applications, features among a data may exhibit more complex smoothness structure. More specifically, features may form an undirected graph structure, where connected features may share some common properties. For example, much biological evidence indicated that genes tend to work in groups if they have similar biological functions [Li and Li (2008)]. This prior knowledge can be encoded by a graph, where in the graph, each node represents a gene and edges denote the regulatory relationships between two associated genes. Recent studies have shown that encoding the structure information as a graph can significantly improve the predictive performance of the model. Given a undirected graph $G \equiv (V, E)$, where $V$ denotes the set of nodes and $E$ denotes the edges. By noting that an open chain is a special example of a graph, we can generalize the fused Lasso penalty to a graph Lasso (GraphL) penalty—*a.k.a.* the $\ell_1$-norm graph Lasso—as follows:

$$\Omega^{\ell_1}_{\text{graphL}}(\mathbf{x}) = \alpha\|\mathbf{x}\|_1 + (1-\alpha)\sum_{(i,j)\in E}|x_i - x_j|. \tag{2.9}$$

In Eq. (2.9), the second term penalizes the difference between coefficients of connected features. As a consequence, coefficients of connected features within the graph tend to be close or identical to each other.

## 2.3 Optimization Methods

Many sparse models in the form of Eq. (2.2) are typically non-smooth and non-differentiable due to the complex sparse-inducing penalties. This fact imposes great challenges to the corresponding optimization algorithms. In the past decade, as sparse models become increasingly popular, extensive research efforts are devoted to developing efficient optimization methods for the sparse models. In the first part of this section, I briefly review two particularly popular first-order methods—proximal gradient descent and accelerated gradient method, which are effective for solving large-scale problems. In the sequel, I introduce the idea of screening for improving the computational efficiency. Moreover, I brief introduce the alternating direction method of multipliers (ADMM) algorithm for solving complex convex optimization problems. In addition, I review the difference of convex functions (DC) programming for handling a class of non-convex problems.

### *2.3.1 Proximal Gradient Descent*

In this section, I briefly review the well-known proximal gradient descent algorithm for solving Problem (2.2). For many sparse models, the loss function $\ell(\cdot)$ is convex and differentiable, while the regularizer $\Omega(\cdot)$ is convex but non-differentiable. The major challenge in developing optimization algorithms for Eq. (2.2) is due to the non-differentiable regularizer $\Omega(\cdot)$.

The proximal gradient descent is an iterative approach. The key idea [Beck and Teboulle (2009); Hastie *et al.* (2015)] is that: in each iteration, we minimize a lo-

cal approximation of $f(\cdot)$ consisting of the non-differentiable part $\Omega(\cdot)$ and a linear approximation of the differentiable part $\ell(\cdot)$. Specifically, in the $k^{th}$ iteration, we optimize $\beta^k$ by the following generalized gradient update:

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \left\{ \ell(\mathbf{x}^k) + \langle \nabla \ell(\mathbf{x}^k), \mathbf{x} - \mathbf{x}^k \rangle + \frac{1}{2t^k} \|\mathbf{x} - \mathbf{x}^k\|^2 + \Omega(\mathbf{x}) \right\}. \tag{2.10}$$

In addition, for a convex function $h$, we can define a proximal map as follows:

$$\mathbf{prox}_h(u) = \arg\min_{\mathbf{v}} \left\{ \frac{1}{2} \|\mathbf{v} - \mathbf{u}\|^2 + h(\mathbf{v}) \right\}. \tag{2.11}$$

Then, it follows that

$$\mathbf{x}^{k+1} = \mathbf{prox}_{t^k \Omega} \left( \mathbf{x}^k - t^k \nabla \ell(\mathbf{x}^k) \right). \tag{2.12}$$

Sufficient conditions [Nesterov (2007)] for the convergence of the update in Eq. (2.12) are as follows:

1. The gradient of the differentiable part $\ell(\cdot)$ is Lipschitz continuous, *i.e.*, for any $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^p$, the following inequality holds:

$$\|\nabla \ell(\mathbf{x}) - \nabla \ell(\mathbf{x}')\|_2 \leq L \|\mathbf{x} - \mathbf{x}'\|_2.$$

2. The step size $t^k$ is a constant that satisfies $t^k \in (0, 1/L]$.

Assume $\mathbf{x}^*$ is an optimal solution, it can be shown that

$$f(\mathbf{x}^k) - f(\mathbf{x}^*) \leq \frac{L \|\mathbf{x}^0 - \mathbf{x}^*\|^2}{2k}. \tag{2.13}$$

Therefore, the above inequality Eq. (2.13) implies that the proximal gradient descent in Eq. (2.12) leads to a convergence rate of $O(1/k)$.

### 2.3.2 Accelerated Gradient Method

When the proximal mapping in Eq. (2.12) can be computed efficiently, the proximal gradient descent approach is a very popular and efficient tool in optimizing the corresponding sparse models, especially for large-scale problems. However, the convergence of proximal gradient descent can be slow for certain objective functions, as the update step may lead to undesirable type of zig-zagging behavior from step-to-step [Hastie *et al.* (2015)]. To this end, in order to improve the convergence property, Nesterov [Nesterov (1983, 2007)] proposed a class of accelerated gradient methods with a convergence rate of $O(1/k^2)$. I summarize the accelerated gradient method in Algorithm 1.

**Algorithm 1** Accelerated Gradient Method

**Input:** A constant $t \in (0, 1/L]$, where $L$ is a Lipschitz constant of $\nabla \ell$.

1: Set $\mathbf{x}^0 = \theta^1 \in \mathbb{R}^p$, $s^1 = 1$, and $k = 1$.

2: **while** termination condition is not satisfied **do**

3: $\quad \mathbf{x}^k = \mathbf{prox}_{t\Omega}(\theta^k - t\nabla \ell(\theta^k))$,

4: $\quad s^{k+1} = \frac{1+\sqrt{1+4(s^k)^2}}{2}$,

5: $\quad \theta^{k+1} = \mathbf{x}^k + \left(\frac{s^k - 1}{s^{k+1}}\right)(\mathbf{x}^k - \mathbf{x}^{k-1})$,

6: $\quad k = k + 1$.

7: **end while**

Let $\mathbf{x}^k$ be generated by Algorithm 1 and $\mathbf{x}^*$ be an optimum. Then, it can be shown that

$$f(\mathbf{x}^k) - f(\mathbf{x}^*) \leq \frac{2L\|\mathbf{x}^0 - \mathbf{x}^*\|^2}{(k+1)^2}. \tag{2.14}$$

It is worth mentioning that, besides the convergence rates, a key difference—that distinguishes the accelerated gradient method from the proximal gradient descent—is that the function values obtained by the former may be increasing, *i.e.*, $f(\mathbf{x}^{k+1})$ may

larger than $f(\mathbf{x}^k)$, while they keep decreasing for the latter one.

In this dissertation research, I develop an approach for optimizing the proximal operator problem associated with the proposed absolute fused Lasso though accelerated gradient method; details is presented in Section 4.3.

### *2.3.3 Screening Rules for Sparse Models*

The idea of *screening* [El Ghaoui *et al.* (2012); Tibshirani *et al.* (2012); Wang *et al.* (2015b)] has been shown to be very promising in boosting the efficiency of large-scale Lasso-related problems. Generally speaking, screening rules aim at quickly identifying the inactive features, which have zero components in the solution. By removing those inactive features from the optimization, screening rules can lead to substantial savings in computational cost and memory usage.

Screening rules are inspired by the Karush-Kuhn-Tucker (KKT) conditions. Recall that Lasso problem presented in Eq. (2.3), its dual is equivalent to:

$$\inf_{\theta} \quad \left\{ \frac{1}{2} \left\| \theta - \frac{\mathbf{y}}{\lambda} \right\|_2^2 : \ |\mathbf{a}_i^T \theta| \leq 1, \ i = 1, 2, \ldots, p \right\}, \tag{2.15}$$

where $\theta$ is a dual variable. Let $\mathbf{x}^*(\lambda)$ and $\theta^*(\lambda)$ be the optimal solution of problems (2.3) and (2.15), respectively. Then the primal optimum and dual optimum are related by the KKT conditions as follows:

$$\mathbf{y} = \mathbf{A}\mathbf{x}^*(\lambda) + \lambda\theta^*(\lambda), \tag{2.16}$$

$$(\theta^*(\lambda))^T \mathbf{a}_i \in \begin{cases} \text{sgn}([\mathbf{x}^*(\lambda)]_i), & \text{if } [\mathbf{x}^*(\lambda)]_i \neq 0, \\ [-1, 1], & \text{if } [\mathbf{x}^*(\lambda)]_i = 0, \end{cases} \tag{2.17}$$

where $[\cdot]_k$ denotes the $k^{th}$ component in the coefficient. As a consequence, KKT conditions in Eq. (2.17) leads to:

$$|(\theta^*(\lambda))^T \mathbf{a}_i| < 1 \Rightarrow [\mathbf{x}^*(\lambda)]_i = 0, \textit{i.e.}, \mathbf{a}_i \text{ is an inactive feature.} \tag{R1}$$

The above rule implies that, those inactive features have zero components in $\mathbf{x}^*$ and thus can be removed from the optimization problem. In addition, inspired by the SAFE rules [El Ghaoui *et al.* (2012)], (R1) can be relaxed as follows:

$$\sup_{\theta \in \mathbf{\Theta}} |\mathbf{a}_i^T \theta| < 1 \Rightarrow [\mathbf{x}^*(\lambda)]_i = 0, \textit{i.e.}, \mathbf{a}_i \text{ is an inactive feature}, \qquad \text{(R1')}$$

where $\Theta$ is a set that contains $\theta^*(\lambda)$.

As a side note, in (R1'), the smaller the region $\Theta$ is, the more accurate the estimation of $\theta^*(\lambda)$—*i.e.*, more inactive features can be identified. A useful consequence of (R1) is that we can find a smallest value of $\lambda$ such that $\mathbf{x}^*(\lambda) = 0$. Indeed, we have [Wang *et al.* (2015b)]:

$$\lambda \geq \lambda_{\max} = \|\mathbf{A}^T \mathbf{y}\|_\infty \Leftrightarrow \mathbf{x}^*(\lambda) = 0. \qquad (2.18)$$

The idea of screening achieves great success in many popular sparse models, e.g., Lasso [Wang *et al.* (2015b)], nonnegative Lasso [Wang and Ye (2014)], group Lasso [Wang *et al.* (2015b,b); Tibshirani *et al.* (2012)], mixed-norm regression [Wang *et al.* (2013)], $\ell_1$-regularized logistic regression Wang *et al.* (2014), sparse-group Lasso [Wang and Ye (2014)], tree-structured group Lasso [Wang and Ye (2015)], and the fused Lasso [Wang *et al.* (2015a)].

In this dissertation, I adopt the enhanced dual polytope projections (EDPP) screening rules for Lasso in Section 4.1 [Wang *et al.* (2015b)] to improve the computational efficiency. The EDPP rules have the best performance to date. In addition, I employ the multi-layer feature reduction (MLFre) screening rules for tree-structured group Lasso in Section 4.2. Experiments demonstrate the speedup gained by screening methods can be several orders of magnitude.

### 2.3.4 Alternating Direction Method of Multipliers

For certain complex sparse-inducing regularizers, we can reformulate the original Problem (2.2) to an equivalent constrained problem. In the sequel, such a problem can be addressed using constrained optimization methods (*e.g.*, the augmented Lagrangian method). The alternating direction method of multipliers (ADMM) [Boyd *et al.* (2011)] algorithm is a variant of the augmented Lagrangian method that performs partial updates for dual variables.

Without loss of generality, in this dissertation, I consider the following optimization problem:

$$\min_{\mathbf{x},\mathbf{z}} f(\mathbf{x}) + g(\mathbf{z}) \tag{2.19}$$

$$s.t.\ \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{c},$$

where $f$ and $g$ are convex, $\mathbf{x} \in \mathbb{R}^p$, $\mathbf{z} \in \mathbb{R}^q$, $\mathbf{A} \in \mathbb{R}^{n\times p}$, $\mathbf{B} \in \mathbb{R}^{n\times q}$, and $\mathbf{c} \in \mathbb{R}^n$. With ADMM, I first reformulate the above problem (2.19) as follows:

$$L_\rho(\mathbf{x}, \mathbf{z}, \mu) = f(\mathbf{x}) + g(\mathbf{z}) + \mu^T(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}) + \frac{\rho}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}\|^2, \tag{2.20}$$

with $\mu$ being the augmented Lagrangian multiplier, and $\rho$ being the non-negative dual update step length. ADMM solves this problem by iteratively minimizing $L_\rho(\mathbf{x}, \mathbf{z}, \mu)$ over $\mathbf{x}$, $\mathbf{z}$ and $\mu$. The update rule for ADMM is given by

$$\mathbf{x}^{k+1} := \arg\min_{\mathbf{x}} L_\rho(\mathbf{x}, \mathbf{z}^k, \mu^k), \tag{2.21}$$

$$\mathbf{z}^{k+1} := \arg\min_{\mathbf{z}} L_\rho(\mathbf{x}^{k+1}, \mathbf{z}, \mu^k), \tag{2.22}$$

$$\mu^{k+1} := \mu^k + \rho(\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z}^{k+1} - \mathbf{c}). \tag{2.23}$$

The ADMM method decomposes a large complex optimization problem into a series of simple subproblems and coordinates the local solutions to the globally optimal. It is worth mentioning that, although ADMM can be very slow to converge to

a high accuracy, oftentimes it can converge to a modest accuracy—which is sufficient enough for many application—within a few tens of iterations.

In my dissertation study, I adopt ADMM to solve the propose two-level structured sparse model in Section 4.4.

### *2.3.5 DC Programming for Non-Convex Optimization*

Sometimes, the sparse-inducing regularizer in Eq. (2.2) can also be non-convex. In this dissertation, I proposed a non-convex absolute fused Lasso penalty in Section 4.3. A key to solve the corresponding optimization problem is though the difference of convex functions (DC) programming [Tao *et al.* (1988); Tao and An (1997); Tao *et al.* (2005)]. I brief review the idea of DC programming in this section.

As an approach that applying convex analysis to non-convex problems, DC programming has been adopted in various non-differentiable non-convex optimization problems. A particular DC program on $\mathbb{R}^p$ takes the form of:

$$f(\mathbf{x}) = f_1(\mathbf{x}) - f_2(\mathbf{x}), \tag{2.24}$$

with $f(\cdot)$ being non-convex on $\mathbb{R}^p$, but $f_1(\cdot)$ and $f_2(\cdot)$ being convex. The algorithm to solve a DC program—which has been introduced in [Tao *et al.* (1988)]—is based on the duality and local optimality conditions. Denote the affine minorization of $f_2(\mathbf{x})$ as $f_2^k(\mathbf{x}) = f_2(\mathbf{x}^k) + \langle \mathbf{x} - \mathbf{x}^k, \partial f_2(\mathbf{x}^k) \rangle$, where $\langle \cdot, \cdot \rangle$ refers to the inner product. A general DC program solves Problem (2.24) by iteratively solving:

$$\min_{\mathbf{x} \in \mathbb{R}^p} f_1(\mathbf{x}) - f_2^k(\mathbf{x}). \tag{2.25}$$

Since $\langle \mathbf{x}^k, \partial f_2(\mathbf{x}^k) \rangle$ is a constant within each iteration, Problem (2.25) is equivalent to:

$$\min_{\mathbf{x} \in \mathbb{R}^p} f_1(\mathbf{x}) - \langle \mathbf{x}, \partial f_2(\mathbf{x}^k) \rangle. \tag{2.26}$$

When problem (2.26) is convex, we can solve it through convex optimization methods.

To sum up, the DC algorithm (DCA) can be summarized as follows: from an appropriate starting point $\mathbf{x}^0$, we iteratively solve Eq. (2.26) until the stopping criterion is satisfied.

In general, DCA cannot guarantee the solution to be the globally optimal, due to the local characteristics and the non-convexity of the original problem. However, it is worth mentioning that, some researchers observed that a DC algorithm converges quite often to a global one [Tao and An (1997)].

## 2.4 Exploiting Label Structure in Multi-Label Learning

In many practical applications, not only features but labels may exhibit some structure, especially in multi-label learning (MLL). In MLL, an instance is associated with multiple targets or labels; for example, text classification and image annotation. In such a case, the learning task of inferring a function from those multiple labeled training data—i.e. predicting multi-dimensional targets—is called multi-target prediction [Waegeman *et al.* (2013)]. More specifically, when the prediction targets are binary, the task is called multi-label classification (MLC) [Zhang and Zhou (2014); Sorower (2010); Tsoumakas and Katakis (2006)]. Formally, suppose there are $m$ target labels, multi-label learning can be phrased as the problem of finding a model $g : \mathbb{R}^p \to \mathbb{Z}_2^m$, where $\mathbb{Z}_k = \{0, 1, \ldots, k-1\}$.

To learning from multi-label data, plenty of algorithms have been proposed in the past decades. We can categorize those algorithms into the following respects [Zhang and Zhou (2014); Sorower (2010)]: (1) simple problem transformation methods, e.g. Label Powerset [Read (2008)], Binary Relevance [Boutell *et al.* (2004)], Calibrated Label Ranking [Fürnkranz *et al.* (2008)]; (2) simple algorithm adaptation methods, e.g. Tree Based Boosting [Schapire and Singer (2000)], Lazy Learning [Spyromitros

*et al.* (2008)], Deep Learning [LeCun *et al.* (2015)]; (3) dimensionality reduction and subspace based methods, e.g. Multi-label Informed Latent Semantic Indexing [Yu *et al.* (2005)], Multi-label Linear Discriminant Analysis [Wang *et al.* (2010)]; (4) ensemble methods, e.g. Random k labelsets [Tsoumakas and Vlahavas (2007)], Random Decision Tree [Zhang *et al.* (2010)]; (5) generative modeling [McCallum (1999); Wang *et al.* (2008)]; and (6) label structure exploitation [Dembszynski *et al.* (2010); Zhu *et al.* (2005)]. In addition to those methods mentioned above, it is worth mentioning that, exploring and utilizing such label structure is potentially beneficial in multi-label learning.

In my dissertation research, annotating gene expression patterns over the entire brain ontology is one of the major tasks in the ADMBA study. More specifically, based on the image features of a gene $\mathbf{x}_i$, we want to associate it with a vector of target labels $\mathbf{y}_i \in \mathbb{Z}_2^m$, where $m$ refers to the number of brain regions (learning tasks). And thus this learning problem is indeed a multi-label classification problem. However, if we simply treat each label (ontology subdivision) separately—we do not make full use of the structural relationships among labels in the learning procedure—it may result in suboptimal predictive performance [Silla Jr and Freitas (2011); Tsoumakas *et al.* (2010)].

To this end, I propose a novel label structure-based two-stage multi-label classification approach, which utilizes the hierarchy structure among labels. The major reasons are: (1) in the atlas, expression patterns of a single gene are recorded based on a hierarchically organized ontology of brain anatomical structures; and (2) it is possible to propagate annotation results to a parent or a child subdivision under a set of systematic rules. Briefly speaking, the proposed learning approach divides the learning process into two stages: in the first stage, a set of interesting tasks are learned individually, and in the second stage, knowledge learned from the first stage will be

utilized to train models as auxiliary features for the remaining tasks. I present more details about this idea in Section 3.3.

# Chapter 3

# ALLEN BRAIN IMAGING – GENE EXPRESSION STUDY

In this chapter, I brief introduce my dissertation research in the Allen brain imaging – gene expression study from three respects. First of all, I present an image feature extraction framework that utilizing SIFT method, sparse coding, and average-/max-pooling in Section 3.1. Next, I introduce my approach that utilizing multi-task sparse logistic regression for multi-class annotation in Section 3.2. Moreover, I propose a novel label structure-based multi-label classification approach in Section 3.3. In the last, I present some experimental results of the proposed methods in Section 3.4.

## 3.1 Proposed Feature Extraction Framework

The problem of annotating gene expression status is essentially an image annotation problem. While for image annotation, how to extracting and characterizing features from images are foundational. Basically, to capture as many details of gene expression over the entire brain ontology, the Allen brain atlas provides numerous spatiotemporal high-resolution ISH images. However, those raw images are not well aligned, as they were taken from different samples and at different spatial slices. This fact makes it challenging to extract features from raw ISH images. To this end, I propose an image feature extraction framework that utilizing SIFT (scale-invariant feature transform) method, sparse coding and different pooling methods. Briefly, I first employ the SIFT approach to detect and describe local image features. Next, I use an augmented sparse coding method to efficiently learn the dictionary from SIFT descriptors of all ISH images and generate patch-level sparse feature representations.

Different pooling methods are utilized to combine patch-level representations to form image-level features, and further generate gene-level representations. A schematic flowchart of the feature extraction framework is shown in Figure 3.1.

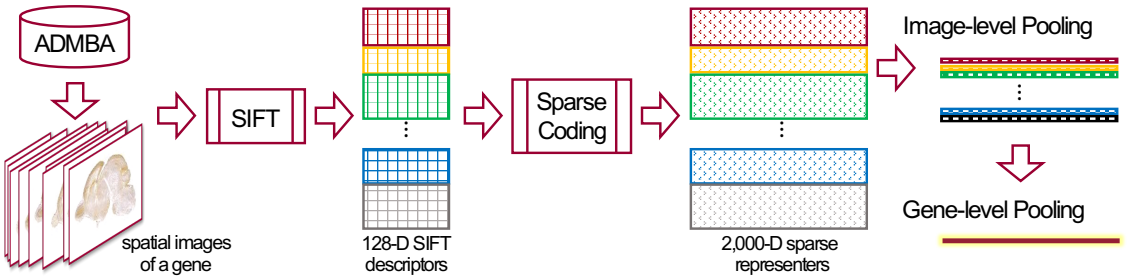



**Figure 3.1:** Schematic flowchart of the feature extraction framework.

### *3.1.1 SIFT for Image-level Feature Extraction*

To detect and describe local image features, I employ the well-known scale-invariant feature transform (SIFT) method in this study. Briefly speaking, the SIFT method first detects multiple localized keypoints (patches) from a raw image, and then transforms those image content into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters. I utilize VLFeat [Vedaldi and Fulkerson (2008)] for SIFT detection and description. As a result, an average of 3,500 patches have been captured for each ISH image, where each patch is represented by a 128-dimensional SIFT descriptor.

### *3.1.2 Sparse Coding for High-Level Feature Construction*

Based on the SIFT descriptors obtained in the previous section, I next apply sparse coding to generalize high-level image patch representations. Sparse coding aims at reconstructing the data vectors through sparse linear combinations of basis vectors and learning a non-orthogonal and over-complete dictionary, which has more

flexibility to represent the data [Olshausen *et al.* (1996); Chen *et al.* (1998); Donoho and Elad (2003)]. It has been applied in many fields such as audio processing and image recognition [Szlam *et al.* (2012)].

Indeed, the sparse coding problem can be formulated as follows:

$$\min_{\mathbf{D},\mathbf{z}^1,\ldots,\mathbf{z}^n} \sum_{i=1}^{n} (\frac{1}{2}\|\mathbf{D}\mathbf{z}^i - \mathbf{a}^i\|_2^2 + \theta\|\mathbf{z}^i\|_1) \tag{3.1}$$

$$s.t.\ \|\mathbf{D}_{\cdot j}\|_2 \leq 1, 1 \leq j \leq p$$

where $\mathbf{A} = [\mathbf{a}^1, \ldots, \mathbf{a}^n] \in \mathbb{R}^{n \times m}$ is the set of SIFT descriptors obtained from image patches, each SIFT descriptor $\mathbf{a}^i \in \mathbb{R}^m$ is a $m$-dimension (here $m = 128$) normalized vector (*i.e.*, with zero mean and unit norm), $\mathbf{D} \in \mathbb{R}^{m \times p}$ is the coding dictionary, $\theta$ is the regularization parameter, and $\mathbf{Z} = [\mathbf{z}^1, \ldots, \mathbf{z}^n] \in \mathbb{R}^{n \times p}$ is the set of sparse feature representations of the original data. In addition, to prevent elements in the dictionary $\mathbf{D}$ from taking arbitrarily large values, the constraint $\mathbf{D}_{\cdot j}, 1 \leq j \leq p$ is preferred to restrict each column of $\mathbf{D}$ to be in a unit ball.

It is worth mentioning that solving the sparse coding problem is computationally expensive, especially when dealing with a large-scale data set and learning a large size of the dictionary. The primary computational cost comes from the updating of sparse codes and the dictionary. To this end, I adopt a new approach, called stochastic coordinate coding (SCC) [Lin *et al.* (2014)] in this study. It has been shown to be much more efficient than existing methods [Lin *et al.* (2014)]. Key ideas of SCC are: (1) alternately update the sparse codes via a few steps of coordinate descent, and (2) update the dictionary via a second order stochastic gradient. In addition, the computational cost of sparse coding can be further reduced, if we just focus on the non-zero components of the sparse codes and the corresponding dictionary columns during the updating procedure.

In this study, the dictionary is learned based on the SIFT descriptors of image

patches from all ISH images. A set of constraint, $\mathbf{z}^i \geq 0$, $1 \leq i \leq n$, are further added to ensure the non-negativity of sparse codes.

In the sequel, to generate image-level features, I adopt the max-pooling operation based on patch-level representations. Max-pooling takes the strongest signal among multiple patches to represent an image, which has shown to be powerful in combining low-level sparse features [Boureau *et al.* (2010)].

### *3.1.3 Gene-level Feature Pooling*

Recall that a specific ISH image is obtained from particular brain spatial coordinates, and thus it cannot present the gene expression pattern over the entire brain ontology. To this end, to describe expression status in all brain regions, I utilize a gene-level feature pooling to combine multiple ISH images of a gene. In this study, both average-pooling and max-pooling are employed to generate gene-level feature representations of gene expression images.

## 3.2 Group Lasso for Multi-Class Annotation

In the Allen study, annotating the detailed categories of gene expression status [see Figure 1.1, (a-c)] is essentially a multi-class classification problem. In this section, I introduce my method for solving the multi-class annotation problem via a multi-task learning (MTL) approach. Briefly speaking, MTL aims at learning these related tasks simultaneously by extracting and utilizing appropriate shared information across tasks [Zhou *et al.* (2011); Evgeniou and Pontil (2007, 2004)]. Multi-task learning has been empirically [Ando and Zhang (2005); Caruana (1997); Bakker and Heskes (2003); Evgeniou *et al.* (2005)] as well as theoretically [Ando and Zhang (2005); Baxter (2000); Ben-David and Schuller (2003); Bakker and Heskes (2003); Baxter (1997)] shown to be promising in terms of predictive performance relative to learning

each task independently. Basically, Multi-task learning is a tool for modeling from a set of related tasks. Thus if the multiple classes are inherently related, it is potentially beneficial to employ MTL method for model construction.

Suppose that there are $k$ classes ($k = 3$ or 4 in this study) in total. I first transform the class representation of a sample to a $k$-tuple vector, where $y_k^i = 1$ if sample $i$ belongs to class $k$ and $y_k^i = 0$ otherwise. Then, the response vector $\mathbf{Y}$ can be written as $\mathbf{Y} = \{\mathbf{y}^i\}_{i=1}^n \in \mathbb{R}^{n\times k}$. In this dissertation study, I employ a $\ell_{2,1}$-norm based structured sparse method together with logistic loss for multi-class classification. More specifically, I employ the following multi-task sparse logistic regression model:

$$\min_{\mathbf{X}} \ell(\mathbf{ZX}, \mathbf{Y}) + \lambda\|\mathbf{X}\|_{2,1}, \tag{3.2}$$

where $\ell(\cdot)$ denotes the logistic loss, $\mathbf{Z} \in \mathbb{R}^{n\times p}$ is the gene-level representations (after patch-level pooling and image-level pooling), $\mathbf{X} \in \mathbb{R}^{p\times k}$, and the $i$-th column of $\mathbf{X}$ refers to the model weight for the $i$-th task (class). The group sparse-inducing regularizer—*i.e.*, $l_{2,1}$-norm penalty on $\mathbf{X}$—leads to grouped sparsity. In other words, it restricts all tasks to share a common set of features during modeling. In this dissertation, the SLEP [Liu *et al.* (2009b)] package is utilized to solve the multi-class learning Problem (3.2).

## 3.3 Label Structure-Based Two-Stage Learning Framework for Multi-Label Annotation

In the Allen brain imaging – gene expression study, annotating gene expression patterns over the entire brain ontology is indeed a multi-label classification problem.

It is worth emphasizing that, the expression status of a single gene are recorded based on a hierarchically organized brain ontology in reference atlas. In addition, in practice, it is possible to propagate annotation to parent or child structures under

a set of systematic rules [Allen Institute for Brain Science (2013)]. Therefore, if we simply treat each label (ontology subdivision) separately—we do not make utilize the structural relationships among labels in the learning procedure—it may result in sub-optimal predictive performance [Silla Jr and Freitas (2011); Tsoumakas *et al.* (2010)]. Alternatively, rather than treating each individual annotation task separately, if we build all prediction models together by utilizing the structural information among labels, the predictive performance can potentially be significantly improved [Silla Jr and Freitas (2011); Tsoumakas *et al.* (2010)].

To this end, I propose a novel label structure-based two-stage multi-label classification approach in this study. It makes full use of the hierarchy structure of labels. A basic idea is of the proposed method is presented in Figure 3.2. Essentially, I divide the learning process into two stages: in the first stage, a set of interesting tasks are learned individually, and in the second stage, knowledge learned from the first stage will be utilized as auxiliary features to train models for the remaining tasks.

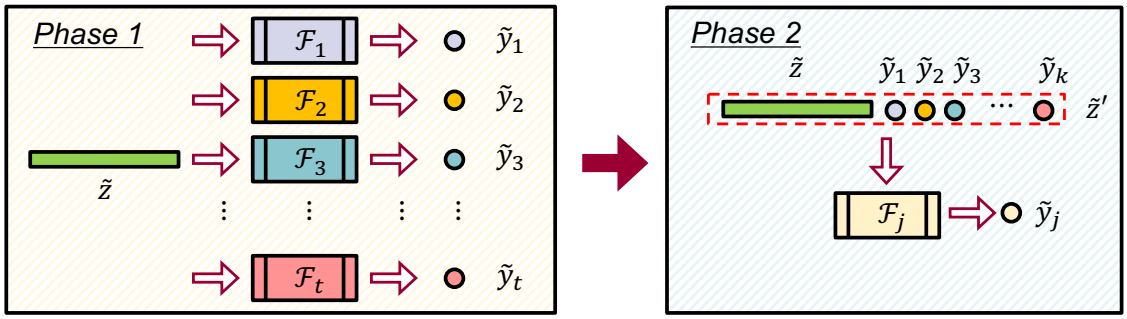



**Figure 3.2:** Two-stage learning framework. $\tilde{z}$ represents a sample, $\mathcal{F}$ is a proper learning function, $\tilde{y}$ is the corresponding learning result, $j \notin \{1, \ldots, t\}$.

Formally, suppose we are given $n$ training data points $\{(\mathbf{z}^i, \mathbf{y}^i)\}_{i=1}^n$, where $\mathbf{z}^i \in \mathbb{R}^p$ is a sample of $p$ features, and $\mathbf{y}^i \in \mathbb{R}^k$ is the corresponding label vector of $k$ tasks. In addition, denote $j \in \{1, \ldots, k\}$ be the $j$-th learning task. Then, I divide the learning procedure into two stages. Specifically, in the first stage, I pre-select $t$ tasks ($t < k$)

that we are interested in, and each of those tasks is learned individually by:

$$\tilde{y}_j = \mathcal{F}_j(\tilde{\mathbf{z}}),\ 1 \leq j \leq t < k, \tag{3.3}$$

where $\mathcal{F}_j(\cdot)$ denotes a learnt model by the $j$-th task, $\tilde{\mathbf{z}} \in \mathbb{R}^p$ is an arbitrary data point, and $\tilde{y}_j \in \mathbb{R}$ is the prediction of $\tilde{\mathbf{z}}$ for the $j$-th task. Note that the order in Eq. (3.3) is just for easy presentation purpose. In the second stage, the learned knowledge in Eq. (3.3) is then used to train the remaining tasks (*i.e.*, $t+1 \leq j \leq k$). More specifically, I augment the feature set by adding the prediction probabilities learnt in the previous stage, *i.e.*, I denote an augmented feature set by $\tilde{\mathbf{z}}' = [\tilde{\mathbf{z}}, (\tilde{y}_1, \ldots, \tilde{y}_t)]$. In the sequel, annotation tasks in the second stage will be performed based on this augmented features.

The tasks in the first learning stage can be considered as the auxiliary tasks in the second stage [Ando and Zhang (2005)]. I propose to consider such a two-stage multi-label learning approach in this study since the tasks are not symmetric due to the hierarchical label structure. With the prediction probabilities from the previous learning stage, I make use of label dependency along with the original image feature representations. Intuitively, if a new learning task is related to some of the tasks in the first stage, then such an approach is expected to achieve better classification accuracy. In my study, since the tasks associated with the bottom-level in the label hierarchy are related to the remaining tasks, the prediction performance is expected to be improved by the two-stage learning approach. This is confirmed in our experiments presented in the next section.

## 3.4 Experiments

In this section, I evaluate the proposed approaches on the Allen developing mouse brain atlas (ADMBA) data sets. More specifically, experiments have been conducted

from the following respects: (1) comparison between sparse coding and bag-of-words, (2) comparison between different multi-class annotation methods, and (3) comparison between annotation with and without brain ontology—i.e., the proposed two-stage learning framework.

### *3.4.1 Experimental Setup*

The gene expression ISH images are obtained from the Allen developing mouse brain atlas. More specifically, to ensure the consistency of brain ontological structure across different developmental stages, I focus the experiments on four embryonic stages: E11.5, E13.5, E15.5, and E18.5. The Allen atlas provides approximately 2,100 genes within the stage and an average of 15$\sim$20 spatially related images are used for each gene to capture the expression information over the entire brain. I use the SIFT method to detect local gene expression and adopt the proposed augmented sparse coding approach to learn sparse feature representations for image patches. Considering the high-resolution of ISH images and the number of regions within the brain ontology, a dictionary of size 2,000 is chosen, *i.e.*, $\mathbf{D} \in \mathbb{R}^{128\times 2000}$. Later, both max-pooling and average-pooling are performed to generate gene-level representations.

To evaluate the effectiveness of the proposed feature extraction approach, I compare it with the state-of-the-art bag-of-words (BoW) method. Specifically, BoW is performed in two different configurations: the first approach—non-spatial BoW, concatenates three BoW representations of SIFT features, where each BoW is learned from a specific scale of the ISH images; the second approach—spatial BoW, further divides the brain sagittally into seven intervals according to the spatial coordinate of each image. As a result, 21 regional BoW representations are built (7 intervals $\times$ 3 scales) through the spatial BoW method [Zeng and Ji (2014)]. In addition, at each scale, a fixed size of 500 clusters (keywords) are constructed from SIFT representation

and an extra dimension is used to count the number of zero descriptors of each patch.

Recall that R-ADMBA uses three different measurements, including pattern, density, and intensity, to evaluate the gene expression status of each brain ontology region. And basically, the annotation tasks can be considered as either binary-class or multi-class classification problem. For the simple binary-class case, the category "undetected" is treated as the negative class, which refers to the scenario that no gene expression activities are detected at the specific brain region, and all remaining categories are treated as the positive class, which means some kind of expression activities have been detected. It is worth mentioning that, at such a binary-class situation, if the annotation metric "pattern" is marked as "undetected", then metrics "density" and "intensity" must be "undetected", and *vice versa*.

In order to balance the class distributions of training sets, random undersampling on the major class is performed for 11 times. To give a benchmark performance, the experiment results of using Support Vector Machine (SVM) classifier [Chang and Lin (2011)] is also reported. In addition, to better describe the classification performance under the circumstances of data imbalance, I adopt the area under the curve (AUC) of a receiver operating characteristic (ROC) curve as the performance measure for binary-class classification. Moreover, both AUC and accuracy are used as the performance measurements for the multi-class case.

### *3.4.2 Comparison between Sparse Coding and Bag-of-Words*

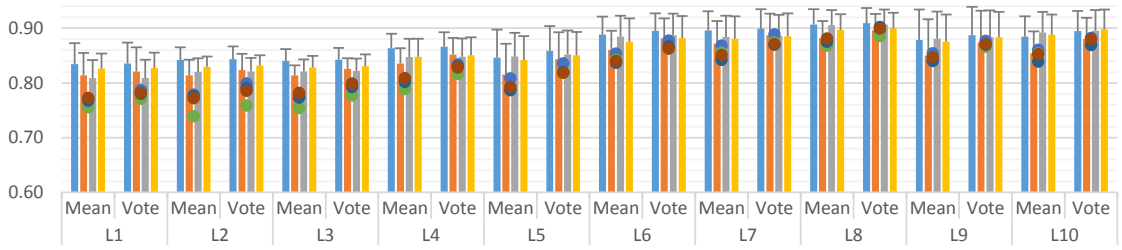
In this series of experiments, I compare the proposed sparse coding (SCC) approach for high-level imaging feature construction with the state-of-the-art bag-of-words (BoW) method. Specifically, raw gene expression ISH images have been processed through the following four methods: (1) SCC_Average, using SCC with a dictionary size of 2,000 to learn image-level representations and adopting average-

pooling to generate gene-level features; (2) SCC_Max, similar to (1) but adopting max-pooling to generate gene-level features; (3) BoW_nonSpatial, generating single BoW representations using all ISH images; (4) BoW_Spatial, generating multiple BoW representations based on images from different spatial coordinates. Here I consider the binary-class situation (*i.e.*, detected vs. undetected), and the original data set is being randomly partitioned into training and testing for each annotation task using a ratio of 4:1. In addition, I adopted the undersampling–majority voting strategy to deal with the imbalanced class distribution. Averaged classification performance in terms of AUC is grouped according to the brain ontological level at different brain developmental stage. Summarization is available in Figure 3.3.

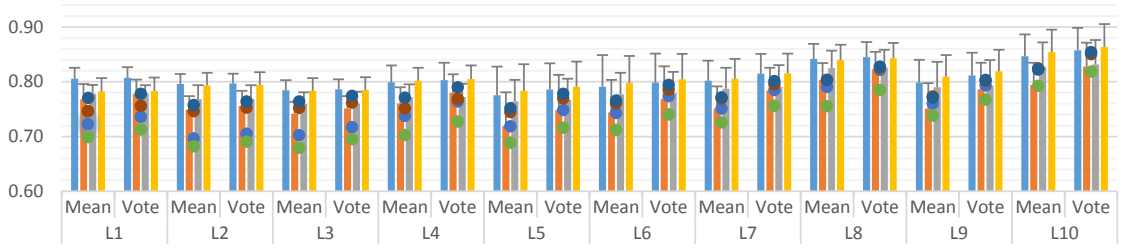
In Figure 3.3, it can be observed that the proposed approach achieves the highest overall predictive performance in terms of AUC. The SCC approaches achieved AUCs of 0.9095, 0.8573, 0.8717 and 0.8903 at mouse brain developmental stages E11.5, E13.5, E15.5, and E18.5, respectively. For the comparison between different types of image representations, SCC_Average achieved the best overall performance among all annotation tasks. Although in some tasks, BoW_Spatial provides competitive performance to SCC_Average, it is worth mentioning that, BoW_Spatial ensembles 21 single dictionaries and contains more than 10,000 features. This implies that BoW_Spatial is far more complex than SCC and involves higher computational costs. Moreover, in comparison with SVM classifiers, the sparse logistic regression classifiers achieve better predictive performance. The above experimental results verify the superiority of our proposed methods.

### *3.4.3 Comparison between Different Multi-Class Annotation Methods*

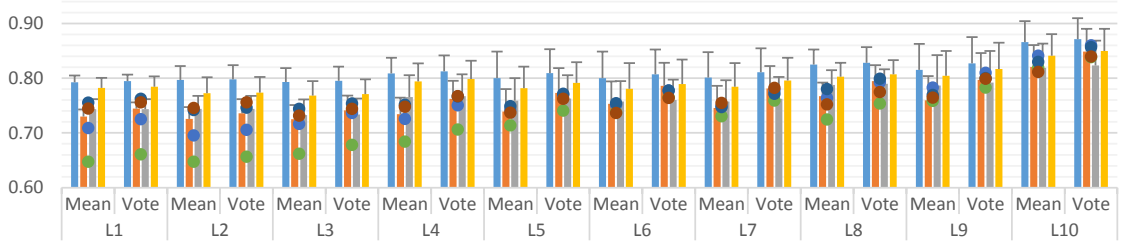
In the following experiments, I evaluate the proposed $\ell_{2,2}$-norm based multi-task sparse logistic regression (mcLR) approach in the multi-class annotation tasks. Based



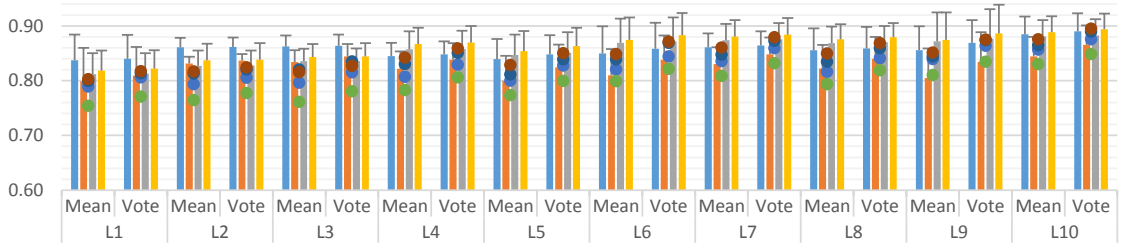

(a) AUC of Annotation Tasks at Different Brain Levels at Stage E11.5.




(b) AUC of Annotation Tasks at Different Brain Levels at Stage E13.5.




(c) AUC of Annotation Tasks at Different Brain Levels at Stage E15.5.




(d) AUC of Annotation Tasks at Different Brain Levels at Stage E18.5.


**Figure 3.3:** Comparison of the proposed approach and bag-of-words method. Each column bar represents the averaged performance of using sparse logistic regression at a specific brain ontological level. Each dot represents the performance of using SVM classifier at a specific brain ontological level. The error bar of each column is the standard deviation of annotation performance within the corresponding brain level. "Mean" group records the average performance of 11 sub-models. "Vote" group records the performance of using majority voting.

**Table 3.1:** Comparison of multi-class annotation methods at stage E11.5.

| | AUC | | | | | | Accuracy(%) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Pattern | | Density | | Intensity | | Pattern | | Density | | Intensity | |
| | SVM | mcLR | SVM | mcLR | SVM | mcLR | SVM | mcLR | SVM | mcLR | SVM | mcLR |
| L5 | 0.708 | 0.713 | 0.678 | 0.734 | 0.688 | 0.689 | 77.26 | 80.38 | 71.52 | 74.93 | 76.98 | 80.90 |
| L6 | 0.711 | 0.729 | 0.700 | 0.733 | 0.715 | 0.710 | 79.29 | 81.78 | 80.68 | 82.97 | 79.93 | 83.57 |
| L7 | 0.731 | 0.732 | 0.684 | 0.739 | 0.712 | 0.709 | 77.69 | 80.43 | 77.34 | 79.88 | 79.33 | 82.54 |
| L8 | 0.711 | 0.734 | 0.704 | 0.735 | 0.736 | 0.726 | 81.61 | 84.35 | 83.40 | 85.46 | 83.98 | 85.80 |
| L9 | 0.640 | 0.666 | 0.688 | 0.699 | 0.693 | 0.698 | 77.02 | 81.10 | 85.40 | 87.16 | 84.84 | 87.04 |
| L10 | — | — | — | — | — | — | — | — | — | — | — | — |

**Table 3.2:** Comparison of multi-class annotation methods at stage E13.5.

| | AUC | | | | | | Accuracy(%) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Pattern | | Density | | Intensity | | Pattern | | Density | | Intensity | |
| | SVM | mcLR | SVM | mcLR | SVM | mcLR | SVM | mcLR | SVM | mcLR | SVM | mcLR |
| L5 | 0.637 | 0.677 | 0.637 | 0.695 | 0.646 | 0.669 | 73.53 | 80.10 | 67.91 | 73.87 | 70.51 | 76.62 |
| L6 | 0.675 | 0.695 | 0.656 | 0.702 | 0.658 | 0.682 | 75.80 | 81.79 | 72.87 | 77.76 | 73.35 | 78.40 |
| L7 | 0.703 | 0.719 | 0.648 | 0.699 | 0.661 | 0.677 | 72.07 | 77.56 | 72.09 | 77.05 | 73.71 | 78.85 |
| L8 | 0.712 | 0.719 | 0.671 | 0.727 | 0.691 | 0.706 | 71.83 | 77.03 | 70.82 | 75.02 | 73.90 | 78.35 |
| L9 | 0.682 | 0.680 | 0.654 | 0.682 | 0.669 | 0.682 | 78.54 | 82.26 | 81.12 | 84.66 | 80.57 | 84.34 |
| L10 | — | — | 0.699 | 0.688 | 0.671 | 0.686 | — | — | 85.36 | 87.48 | 83.22 | 86.00 |

on the results of the previous experiment, I employ the SCC_Average data in this study. In addition, I adopt the multi-class SVM (mcSVM) as the baseline for comparison. In each experiment, 20% of the samples from each class are randomly selected for testing, and the remain samples are used for training. Annotation tasks are included if there are more than 100 samples available for each class ($\sim$2,000 samples in total). Averaged annotation performance at different brain developmental stages in terms of both AUC and accuracy are summarized in Tables 3.1-3.4.

Tables 3.1-3.4 demonstrate that the proposed approach that using sparse logistic regression together with grouped sparsity constraint provides better predictive per-

**Table 3.3:** Comparison of multi-class annotation methods at stage E15.5.

| | AUC | | | | | | Accuracy(%) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Pattern | | Density | | Intensity | | Pattern | | Density | | Intensity | |
| | SVM | mcLR | SVM | mcLR | SVM | mcLR | SVM | mcLR | SVM | mcLR | SVM | mcLR |
| L5 | 0.648 | 0.669 | 0.637 | 0.703 | 0.656 | 0.679 | 80.91 | 86.78 | 70.13 | 74.52 | 72.17 | 77.15 |
| L6 | 0.618 | 0.664 | 0.648 | 0.700 | 0.654 | 0.678 | 79.66 | 83.09 | 76.42 | 80.03 | 76.28 | 80.83 |
| L7 | — | — | 0.645 | 0.702 | 0.667 | 0.687 | — | — | 74.55 | 78.53 | 75.36 | 80.05 |
| L8 | 0.683 | 0.701 | 0.647 | 0.698 | 0.658 | 0.685 | 74.97 | 79.79 | 70.93 | 75.36 | 72.83 | 78.23 |
| L9 | 0.712 | 0.699 | 0.657 | 0.700 | 0.685 | 0.695 | 78.84 | 82.39 | 80.49 | 83.26 | 80.32 | 83.97 |
| L10 | — | — | 0.689 | 0.708 | 0.724 | 0.722 | — | — | 86.16 | 87.61 | 87.03 | 88.26 |

**Table 3.4:** Comparison of multi-class annotation methods at stage E18.5.

| | AUC | | | | | | Accuracy(%) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Pattern | | Density | | Intensity | | Pattern | | Density | | Intensity | |
| | SVM | mcLR | SVM | mcLR | SVM | mcLR | SVM | mcLR | SVM | mcLR | SVM | mcLR |
| L5 | 0.743 | 0.704 | 0.660 | 0.717 | 0.710 | 0.713 | 75.41 | 78.93 | 72.73 | 76.18 | 75.22 | 78.96 |
| L6 | 0.686 | 0.682 | 0.664 | 0.718 | 0.701 | 0.710 | 83.08 | 87.37 | 76.67 | 79.55 | 78.94 | 82.01 |
| L7 | 0.753 | 0.715 | 0.667 | 0.727 | 0.729 | 0.722 | 77.34 | 79.65 | 75.78 | 78.77 | 77.67 | 80.79 |
| L8 | — | — | 0.678 | 0.735 | 0.723 | 0.726 | — | — | 73.79 | 77.48 | 75.89 | 79.58 |
| L9 | 0.745 | 0.717 | 0.681 | 0.728 | 0.744 | 0.731 | 79.49 | 81.42 | 79.17 | 81.56 | 80.59 | 83.01 |
| L10 | 0.720 | 0.702 | 0.693 | 0.710 | 0.750 | 0.740 | 79.38 | 81.45 | 82.94 | 84.96 | 83.52 | 85.56 |

formance in comparison with SVM. Annotation performance of the mcLR approach in both terms of AUC and accuracy are significantly higher than mcSVM at several brain ontology levels. The above experimental results imply that those multiple classes are inherently related, and it is beneficial to learn four (or three) classification models simultaneously by restricting all models to share a common set of features.

### *3.4.4 Comparison between Annotation Performance with/without Brain Ontology*

Recall that in the Allen study, expression status of a single gene are recorded based on a hierarchically organized ontology of brain anatomical structures. It is also

possible to propagate annotation status to a parent or a child subdivision of brain under a set of systematic rules. Therefore, I apply the proposed label structure-based two-stage multi-label learning (SMLL) approach in this study. More specifically, I compare the SMLL method with simple individual approaches—which build models for different tasks independently. At a certain brain developmental stage, around 200 genes are randomly pre-selected as the testing annotation tasks over the brain ontology and the remaining genes are treated as training. For SMLL, 432 tasks (regions) at level 10 (L10) are learned individually in the first stage. Later, the prediction probabilities of L10 tasks will be added into the data set as auxiliary features. In this experiment, I employ the SCC_Average data set and consider the binary-class situation. Preliminary experimental results in terms of both AUC and accuracy are summarized in Tables 3.5 and 3.6.

We can observe from Tables 3.5 and 3.6 that the overall annotation performance achieved by SMLL is higher than individual models. Improvements in terms of AUC and accuracy can be observed at most of the brain ontology levels. This verifies the effectiveness of the proposed label structured-based two-stage multi-label learning approach.

## 3.5 Summary

In this study, I propose an efficient computational approach to perform automated gene expression pattern annotation on mouse brain images. The key information is stored in the form of spatiotemporal in-situ hybridization images. I first employ the SIFT method to construct local image descriptors. I next use sparse coding to efficiently learn the dictionary from SIFT descriptors of all ISH images and generate patch-level sparse feature representations of the images. Different pooling methods are utilized to combine patch-level representations to form image-level features, and

further generate gene-level representations. To discriminate gene expression patterns over each brain area, I employ sparse logistic regression classifier and its multi-task extension to learn models for binary-class and multi-class classification. In addition, random undersampling and majority voting strategies are utilized to deal with imbalanced class distribution inherent within each annotation task. Furthermore, I make full use of the label hierarchy and dependency by developing a novel structure-based multi-label classification approach, which consists of two learning stages. In the first stage, a set of interesting tasks (at the bottom of the label hierarchy) are learned individually, and in the second stage, knowledge learned from the first stage will be utilized to train models for the remaining tasks. I evaluate the proposed approach on the four embryonic mouse developmental stages.

Annotation results show that the adopted sparse coding approach outperforms the bag-of-words method. The proposed method provides favorable classification accuracy on both binary-class and multi-class tasks. Experiment results also show that the structure-based multi-label classification approach can significantly improve the annotation accuracy at all brain ontology levels.

**Table 3.5:** Comparison of annotation performance with/without brain ontology in terms of AUC.

| | E11.5 | | | | E13.5 | | | | E15.5 | | | | E18.5 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | LogisticR | | SVM | | LogisticR | | SVM | | LogisticR | | SVM | | LogisticR | | SVM | |
| | Single | SMLL | Single | SMLL | Single | SMLL | Single | SMLL | Single | SMLL | Single | SMLL | Single | SMLL | Single | SMLL |
| L1 | 0.837 | 0.811 | 0.806 | 0.837 | 0.793 | 0.781 | 0.737 | 0.778 | 0.744 | 0.749 | 0.657 | 0.699 | 0.890 | 0.878 | 0.845 | 0.879 |
| L2 | 0.866 | 0.850 | 0.854 | 0.877 | 0.774 | 0.772 | 0.744 | 0.785 | 0.755 | 0.764 | 0.632 | 0.695 | 0.894 | 0.882 | 0.831 | 0.884 |
| L3 | 0.898 | 0.884 | 0.884 | 0.903 | 0.799 | 0.797 | 0.766 | 0.808 | 0.781 | 0.788 | 0.634 | 0.710 | 0.893 | 0.885 | 0.833 | 0.885 |
| L4 | 0.941 | 0.941 | 0.932 | 0.951 | 0.868 | 0.874 | 0.843 | 0.873 | 0.796 | 0.803 | 0.665 | 0.709 | 0.891 | 0.890 | 0.852 | 0.888 |
| L5 | 0.905 | 0.908 | 0.904 | 0.922 | 0.843 | 0.855 | 0.822 | 0.848 | 0.838 | 0.844 | 0.710 | 0.746 | 0.871 | 0.876 | 0.837 | 0.850 |
| L6 | 0.935 | 0.937 | 0.937 | 0.947 | 0.898 | 0.907 | 0.882 | 0.898 | 0.843 | 0.851 | 0.744 | 0.760 | 0.871 | 0.878 | 0.844 | 0.855 |
| L7 | 0.951 | 0.950 | 0.950 | 0.959 | 0.860 | 0.866 | 0.842 | 0.863 | 0.846 | 0.858 | 0.743 | 0.777 | 0.894 | 0.896 | 0.874 | 0.890 |
| L8 | 0.980 | 0.982 | 0.980 | 0.984 | 0.932 | 0.937 | 0.905 | 0.932 | 0.835 | 0.841 | 0.810 | 0.836 | 0.894 | 0.896 | 0.863 | 0.882 |
| L9 | 0.966 | 0.969 | 0.971 | 0.972 | 0.890 | 0.896 | 0.877 | 0.884 | 0.865 | 0.873 | 0.811 | 0.816 | 0.871 | 0.872 | 0.852 | 0.843 |
| L10 | 0.971 | — | 0.976 | — | 0.906 | — | 0.904 | — | 0.877 | — | 0.837 | — | 0.896 | — | 0.884 | — |

**Table 3.6:** Comparison of annotation performance with/without brain ontology in terms of accuracy.

| | E11.5 | | | | E13.5 | | | | E15.5 | | | | E18.5 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | LogisticR | | SVM | | LogisticR | | SVM | | LogisticR | | SVM | | LogisticR | | SVM | |
| | Single | SMLL | Single | SMLL | Single | SMLL | Single | SMLL | Single | SMLL | Single | SMLL | Single | SMLL | Single | SMLL |
| L1 | 80.89 | 87.32 | 75.00 | 87.86 | 75.33 | 75.33 | 67.11 | 69.90 | 68.42 | 72.86 | 61.68 | 64.47 | 79.40 | 75.00 | 78.14 | 72.86 |
| L2 | 81.67 | 88.69 | 77.14 | 89.52 | 73.57 | 77.08 | 70.07 | 74.12 | 69.85 | 76.32 | 62.06 | 67.76 | 79.56 | 79.15 | 75.54 | 76.32 |
| L3 | 82.59 | 89.77 | 80.30 | 90.71 | 75.07 | 80.85 | 70.50 | 78.57 | 71.68 | 80.54 | 62.64 | 74.24 | 78.44 | 80.96 | 75.54 | 80.54 |
| L4 | 83.67 | 91.89 | 85.00 | 93.83 | 79.32 | 86.65 | 76.50 | 83.79 | 70.30 | 81.06 | 67.72 | 82.10 | 76.31 | 81.16 | 76.85 | 81.06 |
| L5 | 80.94 | 88.50 | 85.31 | 94.67 | 74.54 | 86.44 | 76.39 | 88.27 | 68.91 | 82.60 | 71.69 | 87.22 | 68.47 | 76.78 | 75.12 | 82.60 |
| L6 | 83.48 | 90.22 | 88.70 | 94.95 | 77.59 | 88.09 | 80.62 | 89.37 | 69.91 | 83.02 | 73.57 | 88.97 | 68.60 | 76.44 | 75.30 | 83.02 |
| L7 | 85.22 | 91.06 | 88.24 | 93.33 | 75.88 | 88.41 | 77.32 | 88.48 | 71.56 | 84.72 | 73.92 | 88.87 | 73.20 | 80.66 | 77.74 | 84.72 |
| L8 | 85.86 | 91.98 | 90.41 | 94.80 | 81.74 | 89.51 | 80.38 | 87.53 | 72.78 | 82.29 | 75.64 | 88.11 | 73.64 | 81.02 | 76.52 | 82.29 |
| L9 | 84.09 | 90.73 | 89.19 | 94.76 | 75.44 | 87.34 | 80.83 | 91.55 | 68.31 | 82.58 | 77.89 | 91.58 | 62.76 | 72.09 | 76.35 | 85.77 |
| L10 | 82.41 | — | 88.44 | — | 75.64 | — | 82.24 | — | 70.73 | — | 79.15 | — | 68.41 | — | 78.05 | — |

# Chapter 4

# ADNI IMAGING GENETICS STUDY

In this chapter, I focus on a series of imaging genetic studies that aim to investigate the associations between Alzheimer's disease (AD) phenotypes and genotypes, i.e., how genetic variations—single nucleotide polymorphisms (SNPs)—affect the progression of AD. Those studies are based on the Alzheimer's disease neuroimaging initiative (ADNI) imaging data and whole genome sequence (WGS) data. More specifically, in Section 4.1, I adopt Lasso, as the basic multivariate method, to identify AD-risk SNPs. In the sequel, in Section 4.2, I employ tree-structure group Lasso for the same purpose, by taking advantage of the linkage disequilibrium (LD) information and construct a tree-structure over the SNPs. Moreover, I propose a novel absolute fused Lasso model that can robustly incorporate SNP spatial structure in Section 4.3. To utilize the gene networks over SNPs data, I propose a two-level structured sparse model in Section 4.4. Furthermore, in Section 4.5, I present an approach that utilize convolutional neural networks together with the dropout technique for accurate predicting image-based AD biomarkers. Experiments have been conducted on the ADNI MRI T1 imaging data and WGS data, a suite of selected preliminary experimental results are presented in Section 4.6.

## 4.1 Lasso Method

In the first ADNI imaging genetics study, I employ Lasso regression to identify the most relevant SNPs. Lasso is a simple and basic sparse model for univariate-imaging multivariate-genetic association study. Recall in Eq. (2.3), the non-zero components

in $\mathbf{x}$ corresponding to the relevant features in $\mathbf{A}$. In this study, $\mathbf{A}$ is the processed ADNI WGS SNPs data matrix. As a consequence, Lasso is potentially useful to locate important SNPs that are most relevant to predicting the specific phenotype.

To reveal robust AD-relevant SNPs, I employ the stability selection [Meinshausen and Bühlmann (2010)] method, which is essentially based on subsampling and selection algorithms. Stability selection yields finite sample family-wise error control and markedly improves structure estimation. As it involves solving the Lasso problem many times, such a process can be very time-consuming. To this end, I utilize the enhanced dual polytope projections (EDPP) screening rules [Wang *et al.* (2015b)] to speedup the computation. Lasso together with EDPP screening allows us for the first time to run the compute-intensive model selection procedure to rank causal SNPs that may affect the brain.

### *4.1.1 EDPP Screening Rules for Lasso*

The EDPP screening rules [Wang *et al.* (2015b)] are motivated by the idea of El Ghaoui *et al.* (2012) and Tibshirani *et al.* (2012). Following (R1'), the framework of EDPP screening rules for Lasso can be summarized into the following three steps:

1. Estimate a region $\mathbf{\Theta}$ which contains the dual optimum $\theta^*(\lambda)$.

2. Solve the maximization problem in (R1'), i.e., $\sup_{\theta \in \mathbf{\Theta}} |\mathbf{a}_i^T \theta|$.

3. By plugging in the upper bound we find in the last step, it is straightforward to develop the screening rule based on (R1').

In the above framework, the key is the estimation of the dual optimum, which determines the efficiency of the screening rule. Based on the geometric properties of the dual, EDPP can provide a very accurate estimation of the dual optimum.

In solving Lasso problems, suppose that we are given a sequence of regularization parameter values $\lambda_1 > \lambda_2 > \ldots > \lambda_m$. We first apply EDPP to discard inactive features for the Lasso problem at $\lambda_1$ and compute the optimal solution $\mathbf{x}^*(\lambda_1)$ by solving Lasso on the reduced data matrix. Then, by Eq. (2.16), we can find $\theta^*(\lambda_1)$. In view of (R1'), if we know the dual optimal solution $\theta^*(\lambda_1)$, we can obtain a new screening rule for Problem (2.3) at $\lambda_2$. By repeating the above procedure, we have the sequential version of EDPP screening rules as summarized in Theorem 1.

Let $\lambda_{max} = \|\mathbf{A^T y}\|_\infty$, and let

$$\mathbf{x}_* = \text{argmax}_{\mathbf{x}_i} |\mathbf{x}_i^T \mathbf{y}|, \tag{4.1}$$

$$\mathbf{v}_1(\lambda_0) = \begin{cases} \dfrac{\mathbf{y}}{\lambda_0} - \theta^*(\lambda_0), & \text{if } \lambda_0 \in (0, \lambda_{\max}), \\ \text{sign}(\mathbf{x}_*^T \mathbf{y})\mathbf{x}_*, & \text{if } \lambda_0 = \lambda_{\max}, \end{cases} \tag{4.2}$$

$$\mathbf{v}_2(\lambda, \lambda_0) = \frac{\mathbf{y}}{\lambda} - \theta^*(\lambda_0), \tag{4.3}$$

$$\mathbf{v}_2^\perp(\lambda, \lambda_0) = \mathbf{v}_2(\lambda, \lambda_0) - \frac{\langle \mathbf{v}_1(\lambda_0), \mathbf{v}_2(\lambda, \lambda_0) \rangle}{\|\mathbf{v}_1(\lambda_0)\|_2^2} \mathbf{v}_1(\lambda_0). \tag{4.4}$$

Formally, the sequential version of EDPP can be formulated as follows:

**Theorem 1. EDPP:** *For the Lasso problem, suppose that we are given a sequence of parameter values* $\lambda_{\max} = \lambda_0 > \lambda_1 > \ldots > \lambda_m$. *Then for any integer* $0 \leq k \leq m$, *we have* $[\mathbf{x}^*(\lambda_{k+1})]_i = 0$ *if* $\mathbf{x}^*(\lambda_k)$ *is known and the following holds:*

$$\left| \mathbf{a}_i^T \left( \frac{\mathbf{y} - \mathbf{A}\mathbf{x}^*(\lambda_k)}{\lambda_k} + \frac{1}{2} \mathbf{v}_2^\perp(\lambda_{k+1}, \lambda_k) \right) \right| < 1 - \frac{1}{2} \|\mathbf{v}_2^\perp(\lambda_{k+1}, \lambda_k)\|_2 \|\mathbf{a}_i\|_2. \tag{4.5}$$

The sequential version of the EDPP has several appealing features. First, in a real application, the optimal parameter value of $\lambda$ is typically unknown and needs to be estimated. Second, it can help accelerate the process of stability selection. In our study, I use the DPC package [Wang *et al.* (2015b), `http://dpc-screening.github.io/index.html`] to perform the EDPP screening.

## 4.2 Tree-Structured Group Lasso Method

The $\ell_1$-norm penalty term in the Lasso formulation (2.3) induces sparsity in the coefficients. However, Lasso considers all features equally without any further structural assumptions among them. As mentioned in previous chapters, there are attempts [Liu *et al.* (2011, 2013); Liu (2011)] that utilize LD information together with group Lasso for imaging genetics. However, It is worth mentioning that, with LD information, we can construction a hierarchical tree structure among SNPs as well. In the following study, I incorporate the SNPs' tree structure into the model and apply tree-structured group Lasso (TGL) to identify AD-related SNPs on Chromosome 19 (Chr19).

TGL explicitly incorporates a pre-defined tree structure to characterize the hierarchical relationship among feature set [Liu and Ye (2010)]. For a tree with $d+1$ layers, I denote the set of nodes at depth $i$ by $T_i = \{G_1^i, G_2^i, \ldots, G_{n_i}^i\}$, where $G_j^i$ is the $j$th node at the $i$th layer, $n_0 = 1$, $G_1^0 = \{1, 2, \ldots, p\}$, $p$ is the number of features and $n_i \geq 1$ for $i = 1, 2, \ldots, d$. Each node in the tree denotes a group of features. By convention, for a given index set $G$ and a vector $u$, let $u_G = \{v : v_i = u_i \text{ if } i \in G, v_i = 0 \text{ otherwise}\}$, where $v_i$ is the $i$th component of vector $v$. Then, the TGL problem takes form as follows:

$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|^2 + \lambda \sum_{i=0}^{d} \sum_{j=1}^{n_i} \omega_j^i \|\mathbf{x}_{G_j^i}\|, \tag{4.6}$$

where $\omega_j^i$ is the pre-defined weight for node $G_j^i$.

TGL is a promising technique for revealing the hierarchical sparse patterns among features. To apply TGL to SNPs data, I build the tree structure among SNPs according to the linkage disequilibrium (LD) information and chromosomal locations of SNPs. Briefly speaking, LD refers to the non-independence of alleles at different loci (*i.e.*, positions) in the genome. A widely-used measure of LD between pairs of SNPs is $R^2$ [Pritchard and Przeworski (2001)]. More details regard to the tree construction

are presented in Section 4.6.1.2.

In this study, I apply a similar stability selection framework (refers to Section 4.1) but employ the TGL penalty (2.7) for robust variable selection (*i.e.* identifying AD-risk SNPs). Due to the non-differentiable and high-complexity of TGL, solving such a problem is typically very time-consuming. To this end, I utilize the multi-layer feature reduction (MLFre) rules [Wang and Ye (2015)] for screening. Experiments show that the proposed method is efficient and effective in detecting SNPs that affect AD.

### *4.2.1 MLFre Screening Rules for TGL*

For TGL, let $\phi_j^i(\mathbf{x}) = \|\mathbf{x}_{G_j^i}\|$ and $\partial\phi(\mathbf{0}) = \sum_{i=0}^{d}\sum_{j=1}^{n_i}\omega_j^i\partial\phi_j^i(0)$, where $\phi_j^i(\mathbf{x})$ is the subdifferential [Rockafellar (1970)] of $\phi_j^i$ at $\mathbf{x}$. Let $\mathcal{F} = \theta : \mathbf{A}^T\theta \in \partial\phi(\mathbf{0})$. Then, the dual of TGL can be represented as follows:

$$\sup_{\theta}\left\{\frac{1}{2}\|\mathbf{y}\|^2 - \frac{1}{2}\|\frac{\mathbf{y}}{\lambda} - \theta\|^2 : \theta \in \mathcal{F}\right\}. \tag{4.7}$$

Let $\mathbf{x}^*(\lambda)$ and $\theta^*(\lambda)$ be the optimal solutions of problems (4.6) and (4.7), respectively. The corresponding KKT conditions are:

$$\begin{aligned}\mathbf{y} &= \mathbf{A}\mathbf{x}^*(\lambda) + \lambda\theta^*(\lambda), \\ \mathbf{A}^T\theta^*(\lambda) &\in \sum_{i=0}^{d}\sum_{j=1}^{n_i}\omega_j^i\partial\phi_j^i(\mathbf{x}^*(\lambda)).\end{aligned} \tag{4.8}$$

Let $\mathcal{H}_G = \{u \in \mathbf{R}^p : u_i = 0 \text{ if } i \notin G\}$. Based on the definition of subdifferential [Rockafellar (1970)], we have

$$\omega_j^i\partial\phi_j^i(\mathbf{x}^*(\lambda) = \begin{cases}\left\{\xi \in \mathcal{H}_{G_j^i} : \|\xi\| \leq \omega_j^i\right\}, & \text{if } [\mathbf{x}^*(\lambda)]_{G_j^i} = 0 \\ \omega_j^i[\mathbf{x}^*(\lambda)]_{G_j^i}/\|[\mathbf{x}^*(\lambda)]_{G_j^i}\|, & \text{otherwise.}\end{cases} \tag{4.9}$$

Inspired by KKT conditions in (4.8) and (4.9), the MLFre screening rules take the form of [Wang and Ye (2015)]:

$$\begin{aligned}
&\sup_{\xi}\left\{\|S_j^i(\xi)\| : \xi_{G_j^i} \in \Xi_j^i \supseteq [\mathbf{A}^T\Theta]_{G_j^i}\right\} < \omega_j^i \implies [\mathbf{x}^*(\lambda)] = 0, \text{ if } G_j^i \text{ is a non-leaf node,} \\
&\sup_{\xi}\left\{\|S_j^i(\xi)\| : \xi_{G_j^i} \in [\mathbf{A}^T\Theta]_{G_j^i}\right\} < \omega_j^i \implies [\mathbf{x}^*(\lambda)] = 0, \text{ if } G_j^i \text{ is a leaf node,}
\end{aligned} \tag{4.10}$$

where $[\mathbf{A}^T\Theta]_{G_j^i} = [\mathbf{A}^T\Theta]_{G_j^i} : \lambda \in \Theta$, $\Theta$ is an estimated bounded-ball set containing $\theta^*(\lambda)$ and $\Xi_j^i$ is an estimated set containing $[\mathbf{A}^T\Theta]_{G_j^i}$. Wang and Ye (2015) show that the supremum values on the left-hand sides of (4.10) admit closed-form solutions. For node $G_j^i$ with $[\mathbf{x}^*(\lambda)] = 0$, all features contained by its descendant nodes can be removed from the optimization problem.

## 4.3 Absolute Fused Lasso Method

Tree-structured group Lasso presented in the previous section requires strong prior knowledge among features. In addition, how to build a proper tree structure (distance and linkage functions) and how to choice a certain cutoff level are generally challenging in practice for adopting TGL.

In real-world applications, another scenario that occurs commonly is that the data sets we investigated are of some natural (*e.g.*, spatial or temporal) order; examples include the comparative genomic hybridization data [Tibshirani and Wang (2008)], prostate cancer data [Tibshirani *et al.* (2005)] and neuroimaging data [Yang *et al.* (2012b)]. In such studies, it is often the case that the adjacent samples/features are similar and even identical. Similarly, in genome-wide association studies (GWAS), a causal single-nucleotide polymorphism (SNP) often exhibits high similarity with its nearby SNPs. As a consequence, it is desired to group nearby SNPs together during model selection. In addition, due to the ambiguity choice of reference allele during

genotype coding [Liu *et al.* (2011)], we should group adjacent SNPs if their absolute values are close to each other.

Previous works [Yang *et al.* (2015a); Ye and Liu (2012); Bach *et al.* (2012); Wang *et al.* (2016b)] indicate that utilizing the inherent structural information among the feature is potentially beneficial for model construction as well as interpretation. Thus if the data set exhibits some sequential order, we can potentially incorporate such a prior knowledge into the model to improve performance. Meanwhile, due to the curse of dimensionality in the high-dimensional scenario, identifying the most relevant features that can best explain the outcome is of crucial importance. In ADNI imaging genetics study, the traditional Lasso [Tibshirani (1996)] model is insufficient to produce desired results since it tends to select only one of those highly correlated features [Zou and Hastie (2005)]. There are mainly two approaches in the literature to address the above problem. One approach adopts the fused penalty (e.g., fused Lasso), which can yield a sparse solution in both the coefficients and their successive differences [Tibshirani *et al.* (2005); Tibshirani and Wang (2008); Liu *et al.* (2010)]. However, it does not consider the case that adjacent features are high correlated but with opposite signs. Studies in [Liu *et al.* (2011)] also argue that the fused Lasso is not effective due to the ambiguity choice of coding reference. Another approach utilizes the graph structure among features (*e.g.*, OSCAR) during model construction [Bondell and Reich (2008); Yang *et al.* (2012b); Zhu *et al.* (2013)]. However, such an approach is too general and does not make full use of the specific structure of the genome sequencing data.

To this end, I propose to penalize successive SNPs whose absolute values are close or identical during model learning. More specifically, in my dissertation study, I consider a regularized model which uses a penalty called absolute fused Lasso (AFL) to solve such a problem. The AFL penalty encourages sparsity in the coefficients

as well as their successive differences of absolute values—*i.e.*, local constancy of the coefficient components in absolute value. With AFL, highly similar features can potentially be grouped together even though their signs are different.

It is worth mentioning that the AFL penalty discussed in the next sections is non-convex. And thus it is challenging to develop efficient optimization algorithms. To this end, I employ the difference of convex functions (DC) programming to solve the non-convex optimization problem. At each DC iteration, I adopt the proximal algorithm to efficiently solve the corresponding convex subproblem, which iteratively solves a proximal operator problem; I further use the Barzilai-Borwein (BB) rule for line search to accelerate convergence. One of the major contributions of in my dissertation is to show that such a proximal operator problem regarding AFL can be solved efficiently. More specifically, by exploiting the special structure of the AFL regularizer, I first convert the computation of such a proximal operator to an equivalent optimization problem via a Euclidean projection onto a special polyhedron. I then develop a gradient descent approach based on a novel restart technique by utilizing the optimality condition to efficiently solve the projection problem.

### *4.3.1 The AFL Formulation*

Formally, I consider the following AFL regularization model:

$$\min_{\mathbf{x}\in\mathbb{R}^p} \text{loss}(\mathbf{x}) + \text{afl}(\mathbf{x}), \tag{4.11}$$

where $\text{loss}(\mathbf{x})$ is a convex empirical loss function (*e.g.*, the least squares loss or the logistic loss) and the AFL penalty is defined as:

$$\text{afl}(\mathbf{x}) = \lambda_1\|\mathbf{x}\|_1 + \lambda_2 \sum_{i=1}^{p-1} ||x_i| - |x_{i+1}||, \tag{4.12}$$

where $\lambda_1$ and $\lambda_2$ are non-negative regularization parameters. The second term penalizes differences of successive coefficients' magnitudes and can be considered as a

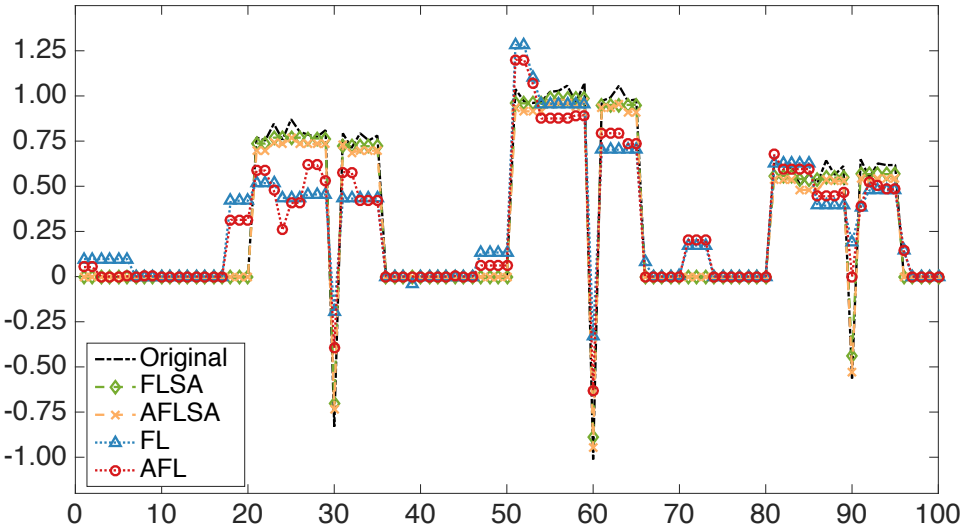



**Figure 4.1:** Comparison of coefficients of the AFL and the fused Lasso (FL) on a simulated data set. FLSA refers to the signal operator of the fused Lasso, AFLSA refers to the signal operator of AFL. When sign flips are part of the true signals, the AFL (red line) provides better recovery of the original signals (black) than the fused Lasso (blue).

grouping penalty. By imposing both the $l_1$ penalty and the grouping penalty, the AFL model can simultaneously identify important features as well as group similar features together (with sign-invariance).

Differing from the fused Lasso that penalizes the $l_1$-norm on successive differences of coefficients, the AFL regularizer encourages the smoothness of adjacent coefficients whose absolute values are close or even identical. As a consequence, strong successive signals can be identified by Eq. (4.11) even when their signs are different. This implies that in general, adopting the AFL penalty is expected to be more effective than the fused Lasso (See an example in Fig. 4.1). Note that in imaging genetics studies, the SNPs data set we obtained through genotype coding is strongly affected by the choice of reference allele. Thus it is insufficient to just penalize the successive differences without considering the absolute values. In Liu *et al.* (2011), the authors use a $l_2$-norm on the absolute difference of adjacent features, and apply coordinate descent to solve the proposed formulation. However, due to the use of $l_2$-norm, the fused

property—*i.e.*, the absolute values of nearby terms tend to be identical—does not hold any more.

In this study, I propose to adopt the DC programming to solve the AFL problem (4.11) and apply a proximal algorithm to solve the sub-problem at each DC iteration. One of the major technical contributions is to develop an efficient solver for computing the proximal operator problem, which is a key building block of the proximal algorithm.

### *4.3.2 DC Programming for Solving the AFL Problem*

The AFL formulation in Eq. (4.11) is non-convex. However, by noting that

$$||x_i| - |x_{i+1}|| = |x_i + x_{i+1}| + |x_i - x_{i+1}| - (|x_i| + |x_{i+1}|),$$

the objective function in Eq. (4.11) can be decomposed into the difference of the following two functions:

$$f_1(\mathbf{x}) = \text{loss}(\mathbf{x}) + \lambda_1\|\mathbf{x}\|_1 + \lambda_2 \sum_{i=1}^{p-1} (|x_i + x_{i+1}| + |x_i - x_{i+1}|),$$

$$f_2(\mathbf{x}) = \lambda_2 \sum_{i=1}^{p-1} (|x_i| + |x_{i+1}|).$$

Therefore, I propose to use the difference of convex functions (DC) programming [Tao *et al.* (1988); Tao and An (1997)] to solve the original AFL problem (4.11).

By linearization of $f_2(\mathbf{x})$, the per-iteration sub-problem of the DC algorithm can be written as:

$$\min_{\mathbf{x}} \text{loss}(\mathbf{x}) - (\mathbf{c}^k)^T\mathbf{x} + \lambda_1\|\mathbf{x}\|_1 + 2\lambda_2 \sum_{i=1}^{p-1} \max(|x_i|, |x_{i+1}|), \tag{4.13}$$

where

$$c_i^k = \lambda_2 d_i \,\text{sgn}(x_i^k) \text{ with}$$
$$d_1 = d_p = 1, d_i = 2, 2 \le i \le p-1 \tag{4.14}$$

and sgn(·) is the signum function (detailed derivation is provided in Appendix A). I summarize the DC algorithm that solves the AFL problem in Algorithm 2. A key building block in this algorithm is how to efficiently optimize the subproblem (4.13). In the following section, I show that Eq. (4.13) can be efficiently solved through a proximal algorithm.

**Algorithm 2** DC algorithm for solving the AFL Problem.

**Input:** data matrix $\mathbf{A} \in \mathbb{R}^{n \times p}$, response vector $\mathbf{y} \in \mathbb{R}^{n \times 1}$,

regularizes $\lambda_1, \lambda_2$, and tolerance $\epsilon$

**Output:** $\mathbf{x}$

1: Initialization: $\mathbf{x}^0 \leftarrow \mathbf{0}, k = 0$

2: **while** $f(\mathbf{x}^k) - f(\mathbf{x}^{k+1}) > \epsilon$ **do**

3: Update $\mathbf{c}^k$ according to Eq. (4.14).

4: Update $\mathbf{x}^{k+1}$ according to Eq. (4.13).

5: $k \leftarrow k + 1$.

6: **end while**

### *4.3.3 The Proximal Algorithm*

In this section, I adopt the proximal gradient descent framework [Wright *et al.* (2009)] to solve the sub-optimization problem (4.13) at each iteration of the DC algorithm. More specifically, Problem (4.13) is equivalent to

$$\min_{\mathbf{x} \in \mathbb{R}^p} h(\mathbf{x}) = l(\mathbf{x}) + m(\mathbf{x}), \tag{4.15}$$

where

$$l(\mathbf{x}) = \text{loss}(\mathbf{x}) - (\mathbf{c}^k)^T \mathbf{x},$$

$$m(\mathbf{x}) = \lambda_1 \|\mathbf{x}\|_1 + 2\lambda_2 \sum_{i=1}^{p-1} \max\left(|x_i|, |x_{i+1}|\right).$$

In the sequel, the proximal algorithm solves problem (4.13) by generating a sequence $\{\mathbf{x}^k\}$ by solving:

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}\in\mathbb{R}^p} \Big\{ l(\mathbf{x}^k) + \langle \nabla l(\mathbf{x}^k), \mathbf{x} - \mathbf{x}^k \rangle + m(\mathbf{x}) + \frac{t^k}{2}\|\mathbf{x} - \mathbf{x}^k\|_2^2 \Big\}, \tag{4.16}$$

where $t^k > 0$ is chosen by some rule introduced below. It is easy to show that (4.16) is equivalent to the following proximal operator problem:

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}\in\mathbb{R}^p} \frac{1}{2}\|\mathbf{x} - \mathbf{u}^k\|^2 + \frac{1}{t^k} m(\mathbf{x}), \tag{4.17}$$

where $\mathbf{u}^k = \mathbf{x}^k - \nabla l(\mathbf{x}^k)/t^k$. In other words, such an algorithm can be viewed as the gradient descent along the direction $-\nabla l(\mathbf{x}^k)$ with the step size $1/t^k$ plus computing the proximal operator problem in (4.17). The pseudo codes of the algorithm are summarized in Algorithm 3.

To guarantee convergence, a line search criterion is adopted to choose an appropriate step size. More specifically, we accept the step size $1/t^k$ if the following inequality holds:

$$h(\mathbf{x}^{k+1}) \leq h(\mathbf{x}^k) - \frac{\sigma}{2} t^k \|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2,$$

where $\sigma \in (0, 1)$ is a constant. To further accelerate the convergence speed of the proximal algorithm, as suggested in studies [Wright *et al.* (2009); Gong *et al.* (2013)], I adopt the Barzilai-Borwein (BB) rule to initialize the line search step size as $1/t^{k,0}$, where

$$t^{k,0} = \frac{\langle \mathbf{a}^k, \mathbf{b}^k \rangle}{\langle \mathbf{a}^k, \mathbf{a}^k \rangle}$$

with $\mathbf{a}^k = \mathbf{x}^k - \mathbf{x}^{k-1}$ and $\mathbf{b}^k = \nabla l(\mathbf{x}^k) - \nabla l(\mathbf{x}^{k-1})$.

Notice that a key step in the proximal algorithm is how to efficiently solve the proximal operator problem in (4.17). In the next section, I introduce an efficient approach to solve Eq. (4.17) by exploiting the special structure of the regularizer.

**Algorithm 3** The Proximal Algorithm.

**Input:** $\mathbf{A}, \mathbf{y}, \lambda_1, \lambda_2$

**Output:** $\mathbf{x}$

Choose $\eta > 1$, $t_{max} > t_{min} > 0$
Initialization: $\mathbf{x}^0, k = 0$
**while** some stopping criterion is not satisfied **do**
    Choose $t^k \in [t_{min}, t_{max}]$
    **while** line search criterion is not satisfied **do**
        Update $\mathbf{x}^{k+1}$ according to Eq. (4.17).
        $t^k \leftarrow \eta t^k$.
    **end while**
    $k \leftarrow k + 1$.
**end while**

#### *4.3.4 Efficient Computation of the Proximal Operator*

For discussion convenience, I absorb $t^k$ into the regularization parameters $\lambda_1$ and $\lambda_2$, and omit the superscript $k$ in Eq. (4.17). Then the proximal operator problem in (4.17) can be further simplified as follows:

$$\pi_{\lambda_2}^{\lambda_1}(\mathbf{u}) = \arg\min_{\mathbf{x}\in\mathbb{R}^p} \Big\{ \frac{1}{2}\|\mathbf{x}-\mathbf{u}\|^2 + \lambda_1\|\mathbf{x}\|_1 + 2\lambda_2 \sum_{i=1}^{p-1} \max\left(|x_i|, |x_{i+1}|\right) \Big\}. \tag{4.18}$$

By applying the procedure discussed in Friedman *et al.* (2007), we have the following theorem:

**Theorem 2.** *For any $\lambda_1, \lambda_2 \geq 0$, we have*

$$\pi_{\lambda_2}^{\lambda_1}(\mathbf{u}) = \mathrm{sgn}(\pi_{\lambda_2}^{0}(\mathbf{u})) \odot \max(|\pi_{\lambda_2}^{0}(\mathbf{u})| - \lambda_1, 0). \tag{4.19}$$

Theorem 2 implies that we can solve Problem (4.18) in two steps: first solve Eq. (4.18) with $\lambda_1 = 0$ and then applying Eq. (4.19) to obtain the final result. In addition, let $\lambda = 2\lambda_2$ and $\lambda_1 = 0$, Eq. (4.18) can be rewritten as:

$$\pi_\lambda(\mathbf{u}) = \arg\min_{\mathbf{x}\in\mathbb{R}^p} \Big\{ \frac{1}{2}\|\mathbf{x}-\mathbf{u}\|^2 + \lambda \sum_{i=1}^{p-1} \max\left(|x_i|, |x_{i+1}|\right) \Big\}. \tag{4.20}$$

In this study, I propose to solve Problem (4.20) efficiently by converting the proximal operator to a Euclidean projection onto a special polyhedron. To perform this transformation, I utilize some important properties of Eq. (4.20) as summarized in Lemma 1, where a detailed proof is provided in Appendix B.

**Lemma 1.** *Let* $\mathbf{x}^* = \pi_\lambda(\mathbf{u})$ *be the optimal solution to* (4.20). $\forall \lambda > 0$, *we have:*

*i)* *if* $u_i \geq 0$, *then* $u_i \geq x_i^* \geq 0$,

*ii)* *if* $u_i < 0$, *then* $u_i \leq x_i^* \leq 0$,

*iii)* $\pi_\lambda(\mathbf{u}) = \mathrm{sgn}(\mathbf{u}) \odot \pi_\lambda(|\mathbf{u}|)$,

*iv)* *if* $|u_i| \geq |u_{i+1}|$, *then* $|x_i^*| \geq |x_{i+1}^*|$,

*v)* *if* $|u_i| < |u_{i+1}|$, *then* $|x_i^*| \leq |x_{i+1}^*|$.

#### *4.3.4.1 Equivalent Euclidean Projection Problem*

Assume $\mathbf{u} \geq 0$, I define a sparse matrix $R \in \mathbb{R}^{(p-1)\times p}$ as follows:

$$R_{ij} = \begin{cases} 1 & u_i < u_{i+1}, j = i \\ 1 & u_i \geq u_{i+1}, j = i+1 \\ -1 & u_i \geq u_{i+1}, j = i \\ -1 & u_i < u_{i+1}, j = i+1 \\ 0 & \text{otherwise.} \end{cases} \tag{4.21}$$

In addition, I denote a vector $\mathbf{w} \in \mathbb{R}^p$ with the $j$-th entry defined as:

$$w_j = \begin{cases} 2 & \sum_i R_{ij} = 2 \\ 0 & \sum_i R_{ij} \leq -1 \\ 1 & \text{otherwise.} \end{cases} \tag{4.22}$$

With Lemma 1 and the above definitions of $R$ and $w$, I next present the following theorem which converts the original proximal operator problem to an equivalent Euclidean projection problem.

**Theorem 3.** *Let* $\mathbf{u} \geq 0$ *and* $\lambda > 0$. *Let*

$$\mathbf{v} = \mathbf{u} - \lambda \mathbf{w} \tag{4.23}$$

*and*

$$P = \{\mathbf{x} | R\mathbf{x} \leq 0, \mathbf{x} \geq 0\}. \tag{4.24}$$

*Define the Euclidean projection of* $\mathbf{v}$ *onto* $P$ *as:*

$$\pi_\lambda^P(\mathbf{v}) = \arg\min_{\mathbf{x} \in P} \frac{1}{2} \|\mathbf{x} - \mathbf{v}\|^2. \tag{4.25}$$

*We have*

$$\pi_\lambda(\mathbf{u}) = \pi_\lambda^P(\mathbf{v}). \tag{4.26}$$

The above theorem implies that, the proximal operator problem in Eq. (4.20) can be solved by solving the Euclidean projection problem in Eq. (4.25). To further simplify, our next theorem shows that, such a Euclidean projection problem can be solved by a simplified problem without the non-negative constraint.

**Theorem 4.** *Let* $\mathbf{u} \geq 0$, $\lambda > 0$,

$$Q = \{\mathbf{x} | R\mathbf{x} \leq 0\}, \tag{4.27}$$

*and*

$$\pi_\lambda^Q(\mathbf{v}) = \arg\min_{\mathbf{x}\in Q} \frac{1}{2}\|\mathbf{x}-\mathbf{v}\|^2. \tag{4.28}$$

*We have*

$$\pi_\lambda^P(\mathbf{v}) = \max(\pi_\lambda^Q(\mathbf{v}), 0). \tag{4.29}$$

Detailed proofs of Theorem 3 and Theorem 4 are provided in Appendices C & D. In the next section, I discuss a restart technique to efficiently solve the Euclidean projection problem in Eq. (4.28).

#### *4.3.4.2 The Restart Technique*

Introducing the dual variable $\mathbf{z} \in \mathbb{R}^{p-1}$ for the inequality constraints in (4.28), we can obtain the Lagrangian in Appendix D.7. The dual problem of Eq. (4.28) is equivalent to

$$\min_{\mathbf{z}\geq 0} \left\{ \phi(\mathbf{z}) = \frac{1}{2}\|R^T\mathbf{z}-\mathbf{v}\|^2 \right\}. \tag{4.30}$$

I propose to solve (4.28) by simultaneously using the information of primal and dual problems. The novelty lies in the usage of the so-called restart technique for fast convergence.

**Optimality Condition and the Support Set**

The proposed restart technique is built on the introduction of the *support set*. Specifically, $\forall \mathbf{z} \geq 0$ and denote $\mathbf{g} = \phi'(\mathbf{z})$, I define the support set as follows:

$$S(\mathbf{z}) = \{i : i \in [1, p-1], z_i = 0, g_i > 0\} \cup \{0, p\}. \tag{4.31}$$

The support set $S(\mathbf{z})$ is motivated by the optimality condition of Problem (4.30), and shall be used for defining a nonlinear and discontinuous mapping from $\mathbf{z}$ to $\mathbf{x}$. $\forall \mathbf{z}^* \geq 0$, it is a minimizer of Eq. (4.30) if and only if $\langle \mathbf{z}-\mathbf{z}^*, \phi'(\mathbf{z}^*)\rangle \geq 0, \forall \mathbf{z} \geq 0$.

From the optimality condition, we can build the relationship between the minimizer and its gradient, as summarized in the following lemma:

**Lemma 2.** *Let* $\mathbf{z}^*$ *be the optimal solution to* (4.30) *and* $\mathbf{g}^* = \phi'(\mathbf{z}^*)$. *We have: i) if* $z_i^* > 0$ *then* $g_i^* = 0$, *and ii) if* $g_i^* > 0$, *then* $z_i^* = 0$.

The matrix $RR^T$ is very special, and it can be shown that its eigenvalues are $2 - 2\cos(i\pi/p), i = 1, 2, \ldots, p-1$, and thus it is positive definite. Note that $RR^T$ is the Hessian of $\phi(\mathbf{z})$, and thus it implies that the minimizer of (4.30) is unique.

**A Nonlinear Mapping $\omega(\cdot)$ from z to x**

Let $s_0 = 0$ denote the smallest entry in $S(\mathbf{z})$, and $s_{|S|} = p$ denote the largest entry in $S(\mathbf{z})$. In addition, let's denote the $j$-th largest entry in the set $S - \{0, p\}$ by $s_j, j = 1, 2, \ldots, |S|-2$. It is clear that $1 \leq s_1$ and $s_{|S|-2} \leq p-1$. With $s_0, s_1, \ldots, s_{|S|-1}$, the indices in $[1 : p]$ can be divided into $|S| - 1$ non-overlapping groups:

$$G_j = \{i : s_{j-1} + 1 \leq i \leq s_j\}, 1 \leq j \leq |S| - 1. \tag{4.32}$$

Let $\mathbf{e} \in \mathbf{R}^p$ be a vector composed of 1's, and $\mathbf{e}_{G_j}$ and $\mathbf{v}_{G_j}$ be the $j$-th group of $\mathbf{e}$ and $\mathbf{v}$ corresponding to the indices in $G_j$, respectively. For discussion convenience, assume $z_0 = z_p = 0$, then we can define the nonlinear mapping $\mathbf{x} = \omega(\mathbf{z})$ based on the support set $S$ as:

$$x_i = \frac{\langle \mathbf{e}_{G_j}, \mathbf{v}_{G_j} \rangle - z_{s_{j-1}} + z_{s_j}}{|G_j|}, i \in G_j, 1 \leq j \leq |S| - 1. \tag{4.33}$$

With Lemma 2 and the definition of support set in Eq. (4.31), it is easy to show that the optimal solution to Problem (4.28) can be exactly recovered by the support set $S(\mathbf{z}^*)$, as stated in the following theorem.

**Theorem 5.** *Let* $\mathbf{z}^*$ *be the minimizer of the dual problem* (4.30), *and* $\mathbf{x}^*$ *be the minimizer of primal problem* (4.28). *Then* $\mathbf{x}^*$ *can be recovered by* $\mathbf{x}^* = \omega(\mathbf{z}^*)$.

**The Restart Technique and Properties**

By introducing the support set $S$, Theorem 5 provides an alternative way to efficiently computing $\mathbf{x}^*$ from $\mathbf{z}^*$. Specifically, we can exactly obtain $\mathbf{x}^* = \omega(\tilde{\mathbf{z}})$, where $\tilde{\mathbf{z}}$ is an appropriate solution with $S(\tilde{\mathbf{z}}) = S(\mathbf{z}^*)$ even if $\tilde{\mathbf{z}} \neq \mathbf{z}^*$. The intuition is that, for a given appropriate solution $\tilde{\mathbf{z}} \neq \mathbf{z}^*$, if $S(\tilde{\mathbf{z}})$ is close to $S(\mathbf{z}^*)$, $\mathbf{x} = \omega(\tilde{\mathbf{z}})$ can be a better approximation than $\tilde{\mathbf{x}} = \mathbf{v} - R^T\tilde{\mathbf{z}}$ for the primal.

I summarize the gradient projection algorithm based on the proposed restart technique in Algorithm 4. Given an iterative solution $\mathbf{z}^k$, I do not perform the gradient projection at the point $\mathbf{z} = \mathbf{z}^k$. Instead, I first compute $\mathbf{x}^k = \omega(\mathbf{z}^k)$. Then, I compute a restart point $\mathbf{z}_0^k$ by $\mathbf{x}^k = \mathbf{v} - R^T\mathbf{z}_0^k$, where $\mathbf{z}_0^k$ can be solved by an equivalent linear system $RR^T\mathbf{z}_0^k = R\mathbf{v} - R\mathbf{x}^k$. Finally, I perform the gradient projection at the restart point $\mathbf{z} = \mathbf{z}_0^k$. Note that $P_0(\mathbf{x})$ is an operator that projects $\mathbf{x}$ onto the non-negative orthant.

**Algorithm 4** Gradient Projection Algorithm with a Restart Technique.

**Input:** $\mathbf{v}, \lambda, R$

**Output:** $\mathbf{z}$

1: Initialization: $\mathbf{z}^0 \leftarrow \mathbf{0}, L = 2 - 2\cos(\pi(p-1)/p)$, $k = 0$;

2: Compute $\mathbf{g}^0 = \phi'(\mathbf{z}^0) = RR^T\mathbf{z}^\mathbf{0} - R\mathbf{v}$;

and set $\mathbf{z}^\mathbf{0} = P_0(\mathbf{z}^\mathbf{0} - \mathbf{g}^0/L)$;

3: **while** not converge **do**

4: Update the support set $S(\mathbf{z}^k)$ according to (4.31);

5: Update $\mathbf{x}^k = \omega(\mathbf{z}^k)$ according to (4.33);

6: Compute $\mathbf{z}_0^k$ as the solution to $RR^T\mathbf{z}_0^k = R\mathbf{v} - R\mathbf{x}^k$;

7: Update $\mathbf{z}^{k+1} = P_0(\mathbf{z}_0^k)$;

8: $k \leftarrow k + 1$;

9: **end while**

#### 4.3.4.3 Discussion

To end this section, I summarize the methodology for solving the proximal operator problem in Eq. (4.18) as follows. I first show that a minimizer of Problem (4.18) can be obtained by applying a soft-thresholding (4.19) on the solution of an alternative optimization problem (4.20). By applying the properties of Eq. (4.20) introduced in Lemma 1 and two variables $R$ and $w$ defined in (4.21) and (4.22), I show that the proximal operator problem in Eq. (4.20) can be convert to an equivalent problem (4.25). In the sequel, I present to optimize an alternative problem (4.28) without the non-negative constraint through eq. (4.29). To solve Problem (4.28), I develop a novel restart technique by introducing the support set in Eq. (4.31) and a nonlinear mapping in Eq. (4.32). I propose to use Algorithm 4 to solve Problem (4.28) for efficient computation.

## 4.4 Sparse Group Lasso with Group Graph Structure Method

In previous sections, I focus on utilizing the SNPs structure on the SNP-level. However, it is worth emphasizing that the interaction mechanisms between multiple SNPs are remaining unclear in real-world. On the contrary, many previous studies were focused on the gene-level. For example, GeneMANIA [Warde-Farley *et al.* (2010)] provides extensive gene networks based on a very large set of functional association data, including protein and genetic interactions, pathways, co-expression, co-localization and protein domain similarity. Therefore, it is potential beneficial to utilize such gene-level network data in imaging genetics researches. To this end, in this section, I consider a two-level structured sparse model, which utilizing gene-level structure information (gene networks) as well as penalizing SNP-level sparsity, for modeling from ADNI data sets.

More specifically, given a centered data matrix $\mathbf{A} \in \mathbb{R}^{n \times p}$ with $n$ observations and $p$ features, and a corresponding label vector $\mathbf{y} \in \mathbb{R}^n$. Denote $\mathbf{g} \in \mathbb{R}^K$ be the gene-level predictors and $\mathbf{s} \in \mathbb{R}^p$ be the SNP-level predictors, respectively. Let $G \equiv (s_K, E)$ be a given undirected graph over genes, where $s_K = \{1, 2, \ldots, k\}$ is a set of nodes, and $E$ is the set of edges. In addition, suppose that the SNP-level predictors $\mathbf{s}$ can be mapped into $K$ gene-level groups, with $p_k$ the number of SNPs in gene $k$, i.e., $\mathbf{s}$ can be represented as $\mathbf{s} = [s_{11} \ldots s_{1p_1} \ \ldots \ s_{k1} \ldots s_{kp_k}]$. I further denote $\mathbf{G}_s = (\mathbf{M}^T\mathbf{g}) \circ \mathbf{s} = [g_1 s_{11} \ g_1 s_{12} \ldots g_1 s_{1p_1} \ g_2 s_{21} \ g_2 s_{22} \ldots g_2 s_{2p_2} \ \ldots \ g_k s_{kp_k}] \in \mathbb{R}^p$, where $\circ$ is the Hadamard product operator, $\mathbf{M} \in \mathbb{R}^{k \times p}$ is a designed mapping matrix, and $g_i, i \in [1, k]$ is the $i$-th element of $\mathbf{g}$. Moreover, let $\mathbf{w_g} \in \mathbb{R}^K$ denote the weight vector corresponding to the gene-level predictor, and $r_{ij}$ denote the weight of the edge between node $g_i$ and $g_j$. Then, in this study, I consider the following optimization problem:

$$\min_{\mathbf{g},\mathbf{s}} \left\{ \ell(\mathbf{y}, \mathbf{x}) + \lambda_1 \|\mathbf{w_g} \circ \mathbf{g}\|_1 + \lambda_2 \sum_{(i,j) \in E} \tau(r_{ij}) |g_i - \text{sgn}(r_{ij}) g_j| + \lambda_3 \|\mathbf{s}\|_1 \right\}, \tag{4.34}$$

where $\tau(r_{ij})$ represent a general monotonically increasing function weight function that enforces a fusion effect between coefficients $g_i$ and $g_j$.

The above problem can be considered as a sparse group Lasso problem together with graph structure on groups, or the sgLasso_gGraph problem. Let $\mathbf{T}$ be the sparse matrix constructed from the edge set $E$ and I ignore the weight vectors, then Problem (4.34) can be simplified as the following matrix form:

$$\min_{\mathbf{g},\mathbf{s}} \ell(\mathbf{y}, \mathbf{G}_s) + \lambda_1 \|\mathbf{g}\|_1 + \lambda_2 \|\mathbf{T}\mathbf{g}\|_1 + \lambda_3 \|\mathbf{s}\|_1. \tag{4.35}$$

### 4.4.1 ADMM for Solving sgLasso_gGraph Problem

Assume $\ell(\cdot)$ to be the least squares loss, then Problem (4.35) can be rewritten as the following constrained optimization problem:

$$\begin{aligned} \min_{\mathbf{g},\mathbf{s},\mathbf{p},\mathbf{q},\mathbf{r}} & \frac{1}{2}\|\mathbf{y}-\mathbf{A}\mathbf{G}_s\|^2+\lambda_1\|\mathbf{p}\|_1+\lambda_2\|\mathbf{q}\|_1+\lambda_3\|\mathbf{r}\|_1 \\ s.t.\ & \mathbf{g}-\mathbf{p}=\mathbf{0},\mathbf{T}\mathbf{g}-\mathbf{q}=\mathbf{0},\mathbf{s}-\mathbf{r}=\mathbf{0}, \end{aligned} \tag{4.36}$$

where $\mathbf{p},\mathbf{q},\mathbf{r}$ are slack variables. Problem (4.36) can be solved by ADMM. The augmented Lagrangian is

$$\begin{aligned} L_\rho(\mathbf{g},\mathbf{s},\mathbf{p},\mathbf{q},\mathbf{r}) =& \frac{1}{2}\|\mathbf{y}-\mathbf{A}\mathbf{G}_s\|^2+\lambda_1\|\mathbf{p}\|_1+\lambda_2\|\mathbf{q}\|_1+\lambda_3\|\mathbf{r}\|_1+ \\ & \mu^T(\mathbf{g}-\mathbf{p})+\nu^T(\mathbf{T}\mathbf{g}-\mathbf{q})+\xi^T(\mathbf{s}-\mathbf{r})+ \\ & \frac{\rho}{2}\|\mathbf{g}-\mathbf{p}\|^2+\frac{\rho}{2}\|\mathbf{T}\mathbf{g}-\mathbf{q}\|^2+\frac{\rho}{2}\|\mathbf{s}-\mathbf{r}\|^2, \end{aligned} \tag{4.37}$$

where $\mu,\nu,\xi$ are augmented Lagrangian multipliers.

**Update g**: In the $(k+1)$-th iteration, $\mathbf{g}^{k+1}$ can be updated by minimizing $L_\rho$ with $\mathbf{s},\mathbf{p},\mathbf{q},\mathbf{r}$ fixed:

$$\begin{aligned} \mathbf{g}^{k+1} =& \arg\min_{\mathbf{g}} \frac{1}{2}\|\mathbf{y}-\mathbf{A}[(\mathbf{M}^T\mathbf{g})\circ\mathbf{s}^k]\|^2+(\mu^k+\mathbf{T}^T\nu^k)^T\mathbf{g}+\frac{\rho}{2}\|\mathbf{g}-\mathbf{p}^k\|^2+ \\ & \frac{\rho}{2}\|\mathbf{T}\mathbf{g}-\mathbf{q}^k\|^2 \\ =& \arg\min_{\mathbf{g}} \frac{1}{2}\|\mathbf{y}-\mathbf{A}Diag(\mathbf{s}^k)\mathbf{M}^T\mathbf{g}\|^2+[(\mu^k+\mathbf{T}^T\nu^k)^T-\rho\mathbf{p}^k-\rho\mathbf{T}^T\mathbf{q}^k]\mathbf{g}+ \\ & \frac{\rho}{2}\mathbf{g}^T(\mathbf{I}+\mathbf{T}^T\mathbf{T})\mathbf{g} \\ =& \arg\min_{\mathbf{g}} \frac{1}{2}\mathbf{g}^T[(\mathbf{B}^k)^T\mathbf{B}^k+\rho(\mathbf{I}+\mathbf{T}^T\mathbf{T})]\mathbf{g}-[\mathbf{y}^T\mathbf{B}^k-(\mu^k+\mathbf{T}^T\nu^k)^T \\ & +\rho(\mathbf{p}^k)^T+\rho(\mathbf{q}^k)^T\mathbf{T}]\mathbf{g} \end{aligned}$$

where $\mathbf{B}^k=\mathbf{A}Diag(\mathbf{s}^k)\mathbf{M}^T$, and $Diag(\cdot)$ is an operation for turning a vector into a diagonal matrix. The above optimization problem is quadratic, and thus the optimal

solution can be obtained by solving the following linear system:

$$\mathbf{F}_g^k \mathbf{g}^{k+1} = \mathbf{b}_g^k, \tag{4.38}$$

where

$$\begin{aligned}
\mathbf{F}_g^k &= (\mathbf{B}^k)^T \mathbf{B}^k + \rho(\mathbf{I} + \mathbf{T}^T \mathbf{T}), \\
\mathbf{b}_g^k &= (\mathbf{B}^k)^T \mathbf{y} - \mu^k - \mathbf{T}^T \nu^k + \rho \mathbf{p}^k + \rho \mathbf{T}^T \mathbf{q}^k.
\end{aligned}$$

Note that $\mathbf{F}_g^k$ is *symmetric positive definite*, and thus Eq. (4.38) can be solved efficiently via the conjugate gradient method.

**Update s**: In the $(k+1)$-th iteration, $\mathbf{s}^{k+1}$ can be updated by minimizing $L_\rho$ with $\mathbf{g}, \mathbf{p}, \mathbf{q}, \mathbf{r}$ fixed:

$$\begin{aligned}
\mathbf{s}^{k+1} &= \arg\min_{\mathbf{s}} \frac{1}{2}\|\mathbf{y} - \mathbf{A}[(\mathbf{M}^T \mathbf{g}^{k+1}) \circ \mathbf{s}]\|^2 + (\xi^k)^T \mathbf{s} + \frac{\rho}{2}\|\mathbf{s} - \mathbf{r}^k\|^2 \\
&= \arg\min_{\mathbf{s}} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\,Diag(\mathbf{M}^T \mathbf{g}^{k+1})\mathbf{s}\|^2 + (\xi^k)^T \mathbf{s} + \frac{\rho}{2}\|\mathbf{s} - \mathbf{r}^k\|^2 \\
&= \arg\min_{\mathbf{s}} \frac{1}{2}\mathbf{s}^T[(\mathbf{C}^k)^T \mathbf{C}^k + \rho\mathbf{I}]\mathbf{s} - [\mathbf{y}^T \mathbf{C}^k - (\xi^k)^T + \rho(\mathbf{r}^k)^T]\mathbf{s},
\end{aligned}$$

where $\mathbf{C}^k = \mathbf{A}\,Diag(\mathbf{M}^T \mathbf{g}^{k+1})$. Similar to update $\mathbf{g}$, the above optimization problem is quadratic, and thus the optimal solution can be obtained by solving the following linear system:

$$\mathbf{F}_s^k \mathbf{s}^{k+1} = \mathbf{b}_s^k, \tag{4.39}$$

where

$$\begin{aligned}
\mathbf{F}_s^k &= \mathbf{C}^T \mathbf{C} + \rho\mathbf{I}, \\
\mathbf{b}_s^k &= \mathbf{C}^T \mathbf{y} - \xi^k + \rho \mathbf{r}^k.
\end{aligned}$$

Note that $\mathbf{F}_s^k$ is *symmetric positive definite*, and thus Eq. (4.39) can be solved efficiently via the conjugate gradient method.

**Update p**: Similarly, $\mathbf{p}^{k+1}$ can be obtained by solving the following problem:

$$\begin{aligned}\mathbf{p}^{k+1} &= \arg\min_{\mathbf{p}} \lambda_1\|\mathbf{p}\|_1 + (\mu^k)^T(\mathbf{g}^{k+1}-\mathbf{p}) + \frac{\rho}{2}\|\mathbf{g}^{k+1}-\mathbf{p}\|^2 \\ &= \arg\min_{\mathbf{p}} \lambda_1\|\mathbf{p}\|_1 - (\mu^k)^T\mathbf{p} + \frac{\rho}{2}\|\mathbf{g}^{k+1}-\mathbf{p}\|^2 \\ &= \arg\min_{\mathbf{p}} \frac{1}{2}\|\mathbf{p}-(\mathbf{g}^{k+1}+\frac{1}{\rho}\mu^k)\|^2 + \frac{\lambda_1}{\rho}\|\mathbf{p}\|_1\end{aligned}$$

The above optimization problem has a closed-firm solution, known as *soft-thresholding*:

$$\mathbf{p}^{k+1} = S_{\lambda_1/\rho}(\mathbf{g}^{k+1}+\frac{1}{\rho}\mu^k), \tag{4.40}$$

where the *soft-thresholding operator* is defined as:

$$S_\lambda(x) = \text{sgn}(x)\max(|x|-\lambda, 0).$$

**Update q**: Similarly, $\mathbf{q}^{k+1}$ can be obtained by solving the following problem:

$$\mathbf{q}^{k+1} = \arg\min_{\mathbf{q}} \lambda_2\|\mathbf{q}\|_1 + (\nu^k)^T(\mathbf{T}\mathbf{g}^{k+1}-\mathbf{q}) + \frac{\rho}{2}\|\mathbf{T}\mathbf{g}^{k+1}-\mathbf{q}\|^2.$$

The closed-form solution of the above problem can be obtained by:

$$\mathbf{q}^{k+1} = S_{\lambda_2/\rho}(\mathbf{T}\mathbf{g}^{k+1}+\frac{1}{\rho}\nu^k). \tag{4.41}$$

**Update r**: Similarly, $\mathbf{r}^{k+1}$ can be obtained by solving the following problem:

$$\mathbf{r}^{k+1} = \arg\min_{\mathbf{r}} \lambda_3\|\mathbf{r}\|_1 + (\xi^k)^T(\mathbf{s}^{k+1}-\mathbf{r}) + \frac{\rho}{2}\|\mathbf{s}^{k+1}-\mathbf{r}\|^2.$$

The closed-form solution of the above problem can be obtained by:

$$\mathbf{r}^{k+1} = S_{\lambda_3/\rho}(\mathbf{s}^{k+1}+\frac{1}{\rho}\xi^k). \tag{4.42}$$

**Update $\mu, \nu, \xi$**: In the $(k+1)$-th iteration, $\mu, \nu, \xi$ are obtained by:

$$\mu^{k+1} = \mu^k + \rho(\mathbf{g}^{k+1}-\mathbf{p}^{k+1}), \tag{4.43}$$

$$\nu^{k+1} = \nu^k + \rho(\mathbf{T}\mathbf{g}^{k+1}-\mathbf{q}^{k+1}), \tag{4.44}$$

$$\xi^{k+1} = \xi^k + \rho(\mathbf{s}^{k+1}-\mathbf{r}^{k+1}). \tag{4.45}$$

I summarize the algorithm for optimizing Problem (4.34) in Algorithm 5.

**Algorithm 5** ADMM for the sgLasso_gGraph Problem

**Input:** $\mathbf{A}, \mathbf{y}, E, \lambda_1, \lambda_2, \lambda_3, \rho$

**Output:** $\mathbf{g}, \mathbf{s}$

1: Initialization: Initialize $\mathbf{g}$ and $\mathbf{s}$.

2: **while** not converge **do**

3: Compute $\mathbf{g}^{k+1}$ according to Eq. (4.38).

4: Compute $\mathbf{s}^{k+1}$ according to Eq. (4.39).

5: Compute $\mathbf{p}^{k+1}$ according to Eq. (4.40).

6: Compute $\mathbf{q}^{k+1}$ according to Eq. (4.41).

7: Compute $\mathbf{r}^{k+1}$ according to Eq. (4.42).

8: Compute $\mu^{k+1}, \nu^{k+1}$ and $\xi^{k+1}$ according to Eqs. (4.43), (4.44) & (4.45).

9: **end while**

## 4.5 Convolutional Neural Networks with Dropout

In the past few years, with the increasing of computing power of modern processors (especially for GPUs), deep learning has attracted increasing attention in academia as well as industry. Generally speaking, deep learning is a concept of the following three aspects [Ranzato (2014); LeCun *et al.* (2015)]: (1) cascade of non-linear transformations, (2) end-to-end learning, and (3) general framework (any hierarchical model is deep). Various deep learning architectures such as convolutional neural networks (CNN) and recurrent neural networks (RNN) have been applied to fields as computer vision, speech recognition, etc., where they have been shown to produce state-of-the-art results on various tasks.

In my dissertation research, I also consider employing deep learning techniques in imaging genetics studies. In genome sequence data, SNPs can be considered as

spatially connected. To utilize such a relationship, we may take advantages of the convolutional layer structure or the recurrent layers. However, oftentimes in real-world applications, there are more than hundreds or thousands of loci available; and thus it makes the RNN model not effective, due to the vanishing gradient problem during backpropagation through time [Hochreiter (1991); LeCun *et al.* (2015)]. In other words, it is in general not possible to learn such a deep RNN model with more than hundreds or thousands time steps. Alternatively, a CNN approach can be beneficial.

A CNN architecture is formed by a stack of distinct layers that transform the input data into an output data through a differentiable function. In this study, an input data instance is the SNPs data of a subject, and the output is a corresponding imaging phenotype (*e.g.* volume of the hippocampus region of the brain). There are several distinct types of layers are commonly used, as presented below:

- **Convolutional layer (CONV)**. This is the core building block of a CNN. Essentially, the convolutional layer's parameters consist of a set of learnable filters. Besides the number of filters, there are two important concepts of the convolutional layer—local connectivity and spatial arrangement. More specifically, the window sizes (height and width) and the stride size.

- **Pooling layer (POOL)**. The pooling layer is another important concept of CNN. It is a form of non-linear down-sampling. Oftentimes, max-pooling or average-pooling is preferred in real-world applications. Same as the convolutional layer, the window sizes and the stride size are the other two important attributes of the pooling layer.

- **ReLU layer (ReLU)**. ReLU refers to the rectified linear units. This is a layer of neurons that applies the non-saturating activation function $f(x) = \max(0, x)$.

- **Fully connected layer (FC)**. Neurons in a fully connected layer have full connections to all activations in the previous layer. Fully connected layer is often used as high-level reasoning in the neural networks.

- **Loss layer**. This layer specifies how the network training penalizes the deviation between predictions and the ground truths, which is typically the last layer in the network. Frequently used loss functions including softmax or sigmoid.

The most common form of a CNN architecture stacks a few CONV-RELU layers, follows them with POOL layers, and repeats this pattern until the raw input has been merged spatially to a small size. Oftentimes, a CNN architecture takes the following pattern:

$$INPUT \rightarrow [[CONV \rightarrow ReLU] \times N \rightarrow POOL?] \times M \rightarrow [FC \rightarrow ReLU] \times K \rightarrow FC,$$

where the $\times$ indicates repetition, $POOL?$ indicates an optional pooling layer, $N \geq 0$ (and usually $N \leq 3$), $M \geq 0$, and $K \geq 0$ (and usually $K < 3$).

Generally speaking, larger neural networks typically work better than smaller neural networks. However, it is also easier to get overfit with larger networks—*i.e.*, models will have relative low predictive performance on the testing than training data. To this end, Srivastava *et al.* (2014) proposed a simple but effective dropout approach to prevent overfitting. While training, dropout is implemented by only keeping a neuron active with some probability $p$ (a hyperparameter), or setting it to zero otherwise. In CNN, a dropout layer is often applied between fully connected layers.

In my dissertation study, I employ the following convolutional neural networks as presented in Figure 4.2, for ADNI imaging genetics. More specifically, the input of the proposed CNN model is a one-dimensional vector of SNP sequence. The input layer is followed by a convolutional layer together with ReLU as the activation function.

Then the max-pooling operation is performed on each filter of the CONV layer. After flattening, I adopt two fully connected layers to transform multiple neurons to a single one to represent the final output (*i.e.* an imaging phenotype). In addition, the ReLU is used as the activation function for the first FC layer, and a dropout technique is adopted at the first FC layer to alleviate overfitting during the training epochs.

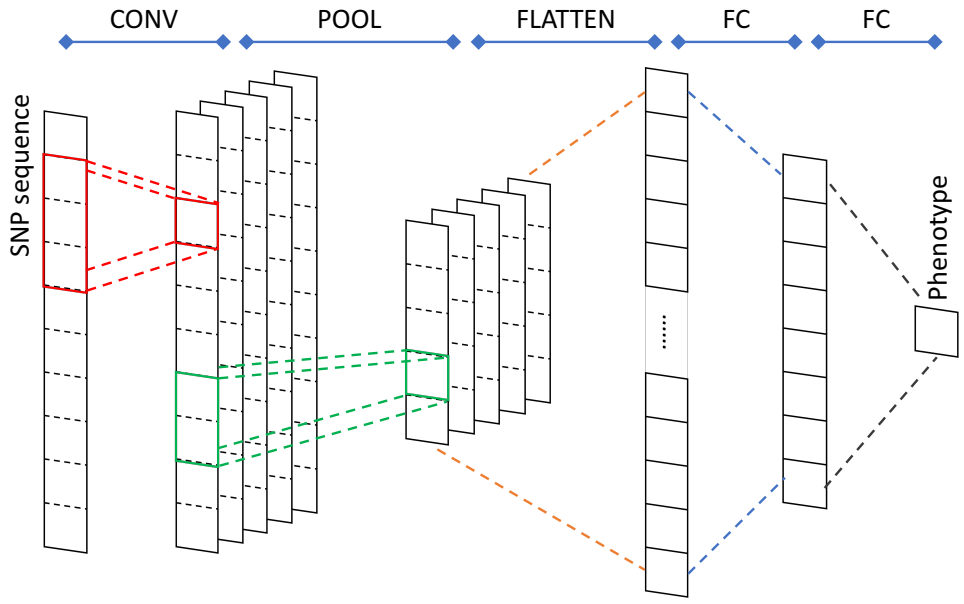



**Figure 4.2:** Architecture of the proposed CNN model. CONV–(w=3,s=1,n=5), POOL–(w=2,s=2), FC_1=256, FC_2=1, DROP_p=0.5.

It is worth emphasizing that, although deep learning approaches can produce state-of-the-art results on tasks such as prediction, its internal mechanisms are remaining unclear to date. In addition, the CNN approach is not capable of identifying AD-risk SNPs in the ADNI imaging genetics study.

## 4.6 Experiments

In this section, I evaluate the proposed approaches on the Alzheimer's disease neuroimaging initiative whole genome sequence data and T1 MRI data. More specifically, I first introduce the data processing procedure of the experiments, including SNPs

data processing, tree-structure construction, and gene-level networks extraction. In the next two sub-sections, I present the experimental results of using Lasso together with EDPP screening rules, as well using tree-structured group Lasso together with MLFre screening rules, for fast identify AD-risk genetic factors. Next, I show the experimental results of the proposed absolute fused Lasso method. In the sequel, the proposed two-level structured sparse method is verified on two sets of selected gene networks. Moreover, I present some preliminary results of adopting CNN for ADNI imaging genetics study. In the last, I compare different structured sparse methods on a set of selected SNPs.

### *4.6.1 Data Processing*

#### *4.6.1.1 Whole genome sequence data*

The ADNI WGS data in this study contains 1,319 subjects, including 327 healthy controls (HC), 249 AD patients, 41 participants with mild cognitive impairment (MCI), 220 early MCI (EMCI) patients, 419 late MCI (LMCI) patients, and 63 patients with significant memory concerns (SMC). For SNPs data, I performed standard quality control in PLINK [Purcell *et al.* (2007)]. Specifically, SNPs were removed with minor allele frequency (MAF) $< 5\%$, missingness $> 5\%$, and deviations from Hardy-Weinberg Equilibrium $P < 5 \times 10^{-7}$ . Genotype imputation was performed by MaCH [Li *et al.* (2010)], which is a Markov chain based haplotyper that can resolve long haplotypes or infer missing genotypes in samples of unrelated individuals. In addition, I apply several filters on the imputed data, including: RSQ (estimated $R^2$, specific to each SNP) $> 0.5$, FREQ1 (frequency for reference Allele 1) $> 1\%$ and FREQ1 $< 99\%$. As a consequence, I obtained a dataset with 1,319 subjects with 6,566,154 SNPs from the entire genome, in which 155,357 SNPs are available on Chromosome

19. Note that the genotype values are sometimes not a discrete number in the set $\{0, 1, 2\}$ since those values are imputed, and the algorithm incorporates uncertainty to the imputed values. In other words, if it's not sure if a subject has 1 or 2 copies of an allele it will make that genotype 1.5. However, such a number must be in the interval $[0, 2]$.

Volumes of key brain regions, including the hippocampus (HIPP) and entorhinal cortex (EC), have been selected as the neuroimaging phenotypes in this study (*i.e.* outcomes). Those values were extracted from subject's T1 MRI data using Freesurfer [Reuter *et al.* (2012)].

#### *4.6.1.2 Tree structure over SNPs*

The hierarchical tree structure among SNPs is built by linkage disequilibrium (LD) or statistical correlations among variants that occur in small windows across the genome. In this study, I use a reference dataset from HapMap release #27 [The International HapMap Consortium (2003)] to build the tree structure of the target dataset. Adjacent SNPs in the reference dataset are grouped together if their pairwise $R^2$ is nonzero. After alignment, I finally obtain 1,063 groups in the target dataset, which serve as the first layer except for the root. Similarly, I then choose two thresholds of $R^2$—0.01 and 0.1 respectively—to group adjacent SNPs if their pairwise $R^2$ values are greater or equal to these thresholds. This results in 5,113 groups in the second layer and 14,883 groups in the third layer respectively. The last layer, as the layer of leaf nodes, contains each single SNP. As a result, a tree structure was constructed with five layers (including the root).

#### 4.6.1.3 *Candidate AD genes and gene networks*

In later studies, I also focus on Alzheimer's disease genetic risk factors (both genes and SNPs) on the 19th chromosome of the human genome. Specifically, at gene-level, ten candidate AD risk genes are pre-selected according to AlzGene (`http://www.alzgene.org/chromo.asp?c=19`), including *LDLR, GAPDHS, BCAM, PVRL2, TOMM40, APOE, APOC1, APOC4, EXOC3L2*, and *CD33*. Positions of those pre-selected genes are shown in Figure 4.3.

The above ten genes have been marked as the most strongly associated genes with

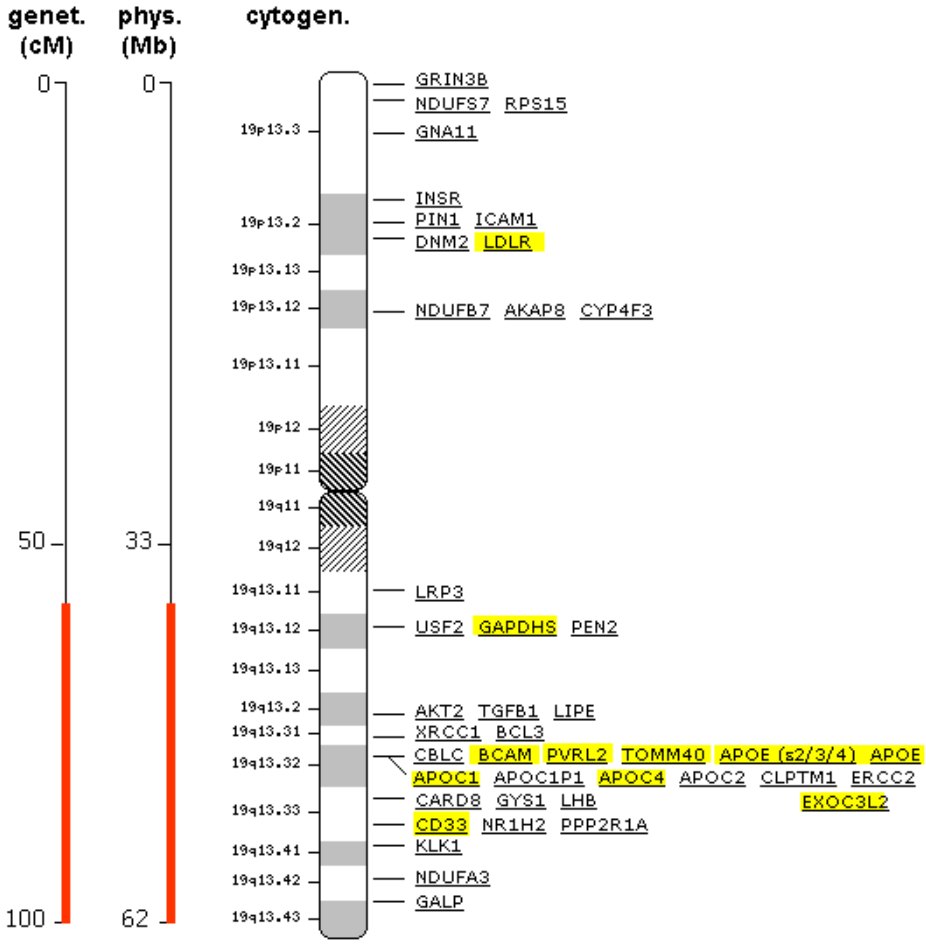



**Figure 4.3:** Candidate AD genes on Chromosome 19 (marked as yellow). Figure adapted from: `http://www.alzgene.org/chromo.asp?c=19`

Alzheimer's disease on Chromosome 19 (Chr19). In AlzGene, top associated genes are ranked based on genetic variants with the best overall HuGENet/Venice grades [Ioannidis *et al.* (2008)]. Specifically, for genes with identical grades, ranking is based on p-value; for genes with identical grade & p-value, ranking is based on effect size.

To explore gene networks, I utilized GeneMANIA [`http://genemania.org/`, Warde-Farley *et al.* (2010)]. Given a set of input genes, GeneMANIA finds gene networks (within given genes as well as other related genes) based on a very large set of functional association data, including protein and genetic interactions, pathways, co-expression, co-localization and protein domain similarity. GeneMANIA stands for *Multiple Association Network Integration Algorithm*. It mainly consists of two parts: 1) a linear regression-based algorithm that calculates a single composite functional association network from multiple data sources; and 2) a label propagation algorithm for predicting gene function given the composite functional association network. More specifically, I use the following two configurations to generate gene networks in my dissertation study:

1. **Gene network within 10 selected AD-related genes in Chr19**.
   The aforementioned ten pre-selected AD risk genes on Chromosome 19 are utilized as input genes for GeneMANIA. For network exploration, I do not enroll new genes; that is, I extract gene network within those ten pre-selected genes. The gene ontology weighting is based on the biological process. A visualization of this gene networks is shown in Figure 4.4.

2. **Extended gene network based on 10 selected Chr19 AD-related genes**.
   Similar to 1, but I enroll ten additional genes for network exploration. This results in totally 20 genes in the network. A visualization of this gene networks is shown in Figure 4.5. Note that, the additional genes are selected based on

their relations with input genes and those genes are not necessary located on Chromosome 19.

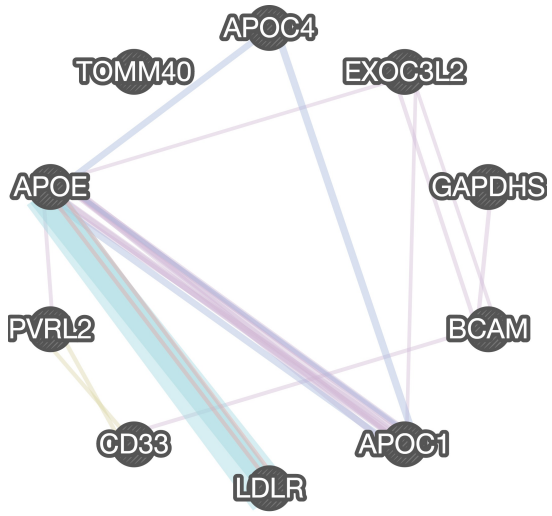



**Figure 4.4:** Network within 10 selected AD-related genes on the 19th Chromosome. Additional network information is available in Appendix E.

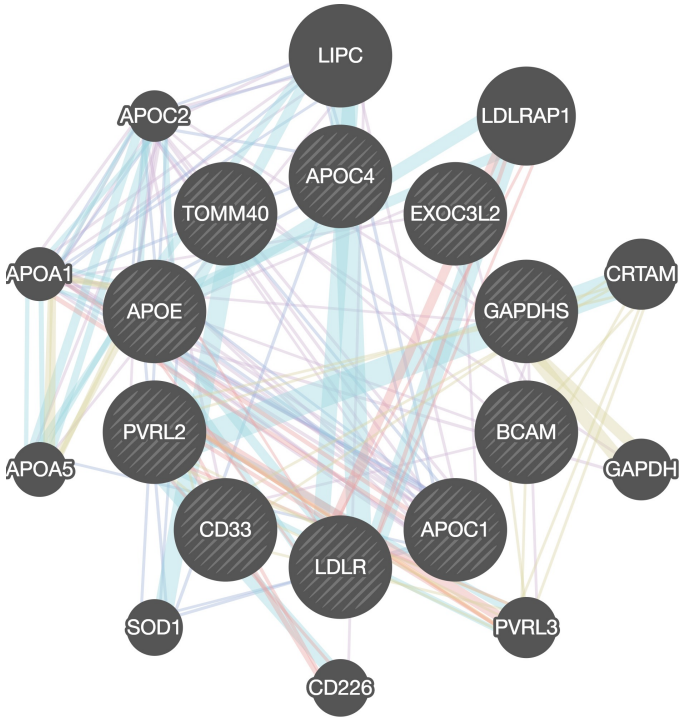



**Figure 4.5:** Extended gene network based on 10 selected Chromosome 19 AD-related genes. Additional network information is available in Appendix E

### 4.6.2 Lasso for ADNI Imaging Genetics

#### 4.6.2.1 Comparison of computational efficiency with and without screening rule

In the first series of experiments, I compare the computational efficiency of Lasso with and without the EDPP screening rule. Specifically, I fix the number of samples and vary the number of features from 0.1 million to 1 million SNPsthat are randomly selectedwith a step size of 0.1 million. The baseline hippocampal volume is chosen be the response vector. For each sub-dataset, I solve a series of Lasso problems at a sequence of 100 parameter values equally spaced in the logarithmic scale from 1.0 to 0.05. The running times are summarized in Figure 4.6.

Figure 4.6 demonstrates that the Lasso solver [Liu *et al.* (2009b)] equipped with EDPP screening rules—i.e., EDPP+Solver—gains a speedup about 406× compared to the solver without screening. In addition, if we double the dimension of the features, the run time of the solver without screening also doubles. However, the run time of the solver with EDPP screening rule only increases slightly in the same situation—which is mainly due to the screening part. This experimental result implies that the EDPP screening rules is a promising approach to facilitate the Lasso solver in dealing with extremely high dimensional data.

#### 4.6.2.2 Models selection results through stability selection

In this experiment, I explore the imaging genetics association between imaging phenotypes and SNPs from the entire ADNI WGS SNP data set. For two brain regions, the entorhinal cortex (EC) and hippocampus (HIPP), I chose the volume at baseline, and volume changes over a 24-month interval as outcomes. I employ stability selection [Meinshausen and Bühlmann (2010)] to obtain the risk SNPs. For each outcome, I perform 100 simulations. In each simulation, I first subsample half

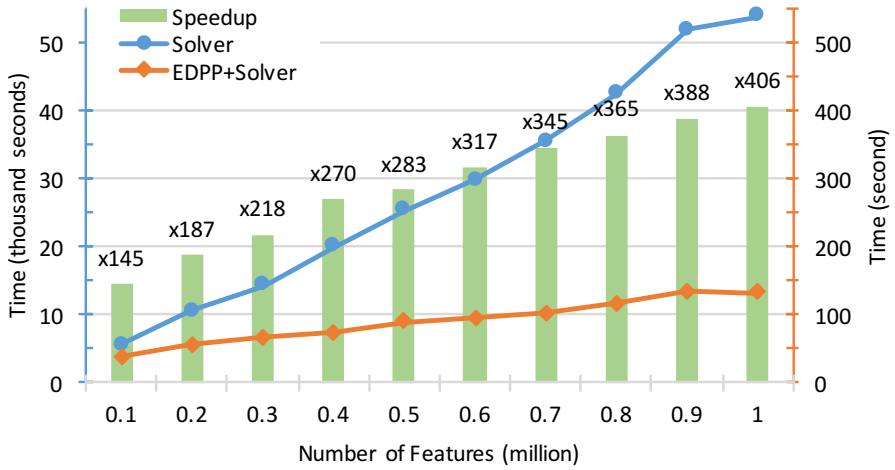



**Figure 4.6:** Comparison of Lasso with and without the EDPP screening rules. Run times in units of kiloseconds are reported for each Solver (Lasso), and in units of seconds for EDPP+Solver.

of the samples from the original data, and then I incorporate EDPP with the solver for Lasso, to solve the Lasso problems at a sequence of 100 parameter values equally spaced on the logarithmic scale of $\lambda/\lambda_{max}$ from 1.0 to 0.05. The selection probabilities for each SNP are recorded and I present the top 10 selected SNPs for each outcome in Table 4.1 and Table 4.2.

In addition, it is worth mentioning that this work allows us for the first time to run the compute-intensive model selection procedure—stability selection, to rank SNPs that may affect the brain and AD risk.

### *4.6.3 Tree-Structured Group Lasso for ADNI Imaging Genetics*

In the following study, I utilize the hierarchical tree structure among the SNPs on the 19th chromosome. Generally speaking, SNPs identified by association analysis or feature selection can be considered as candidate AD-risk factors. Similarly, in this study, I adopted stability selection to rank potential AD risk SNPs by their selection frequencies. Four brain imaging phenotypes from the ADNI MRI data have been chosen as the responses, including volumes of the left entorhinal cortex (LEH), left

**Table 4.1:** Top 10 SNPs associated with baseline volumes selected by Lasso models.

| | EC baseline | | HIPP baseline | |
|---|---|---|---|---|
| | RS_ID | Gene | RS_ID | Gene |
| Rank 1 | rs201890142 | RIMS1 | rs12412466 | PPA1 |
| Rank 2 | 19:15136345 | unknown | rs429358 | APOE |
| Rank 3 | rs6672189 | unknown | rs10831576 | GALNT18 |
| Rank 4 | rs429358 | APOE | rs151073945 | unknown |
| Rank 5 | rs369756382 | ANKRD36C | rs34173062 | MAF1 |
| Rank 6 | rs199536016 | LOC442028 | rs71573413 | unknown |
| Rank 7 | rs200710055 | LOC442028 | rs4825209 | unknown |
| Rank 8 | 1:142545571 | unknown | rs4973360 | unknown |
| Rank 9 | rs76403280 | GPC6 | rs35055545 | OR11H4 |
| Rank 10 | rs202036446 | unknown | rs2343398 | BAI3 |

**Table 4.2:** Top 10 SNPs associated with volume changes selected by Lasso models.

| | EC baseline | | HIPP baseline | |
|---|---|---|---|---|
| | RS_ID | Gene | RS_ID | Gene |
| Rank 1 | rs1317198 | unknown | rs11636690 | NIPA1 |
| Rank 2 | rs1149952 | unknown | rs74977559 | BACE2 |
| Rank 3 | rs146156795 | unknown | rs79543088 | unknown |
| Rank 4 | rs2530339 | GFRA1 | rs7303977 | CACNA1C |
| Rank 5 | rs2912047 | LOC100507530 | rs6605518 | unknown |
| Rank 6 | rs12581794 | unknown | rs34794713 | unknown |
| Rank 7 | rs16946521 | VAT1L | rs9518474 | ITGBL1 |
| Rank 8 | rs9845573 | unknown | rs7889210 | DHRSX |
| Rank 9 | rs4308363 | SORCS2 | rs12646029 | LOC101928478 |
| Rank 10 | rs17502999 | FGF14 | rs149287207 | CLCN3 |

hippocampus (LHP), right entorhinal cortex (REH), and right hippocampus (RHP). For each response, I randomly subsample half of the subjects for 100 times and run TGL equipped with MLFre screening rules on each subsampled data along a sequence of 100 parameter values equally spaced on the linear scale from 1.0 to 0.5.

In principle, SNPs identified by association analysis or feature selection may be-

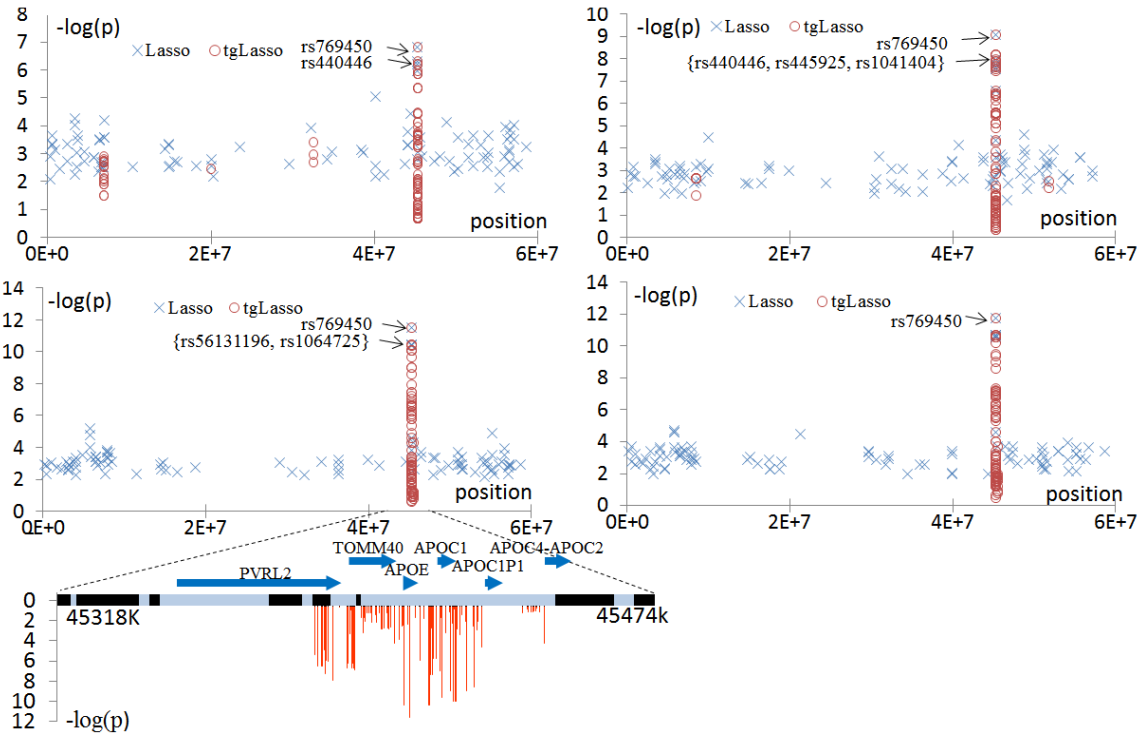



**Figure 4.7:** Top 100 SNPs selected by Lasso and tree-structured group Lasso. Upper left: LEH. Lower left: LHP. Upper right: REH. Lower right: RHP. The horizontal axis is mapped to chromosome position, and the vertical axis is log(p). For LHP, top SNPs for TGL are plotted on a more detailed scale. SNP groups in the second layer of our tree structure are plotted as blocks on the chromosome.

come candidates to be AD-risk factors. I rank the SNPs by their selection frequencies. Figure 4.7 shows the negative of logarithmic of p-values of the top selected SNPs (100 SNPs for each method) together with their position on Chromosome 19. We can observe that SNPs selected by Lasso models are spread over a large region in Chr19. On the contrary, most SNPs selected by TGL models are clustered in a few small chromosome regions. Figure 4.7 also shows the chromosomal regionwhere most top SNPs resideon a more detailed scale for LHP tasks. This scaled region points to several genes, e.g., APOE and TOMM40, which are already repeatedly implicated in AD-risk or risk for other neuropsychiatric disorders [Bertram *et al.* (2007)]. In addition, as shown in the detailed plot of the LHP result in Figure 4.7, this region

consists of SNPs with low p-values and SNPs with high p-values that are distributed across different layers of the tree structure. That is, the TGL approach enhances the power to detect genomic regions that associated AD-related brain measures by a polygenic model.

Moreover, I also show the 39 SNPs that are common in the top 50 lists for all four responses in Table 4.3 and Table 4.4. Encouragingly, the APOE gene is among the top selected genes. This is consistent with the prior studies that indicate that APOE genotype is associated with the volumes of the hippocampus and entorhinal cortex in older adults [Schuff *et al.* (2009); Juottonen *et al.* (1998)].

### *4.6.4 Absolute Fused Lasso for ADNI Imaging Genetics*

To solve the proposed AFL problem, we can adopte the framework of alternating direction method of multipliers (ADMM) or the proximal gradient descent. The ADMM solver can be developed following previous work [Yang *et al.* (2012b)]. However, A major drawback of ADMM is that it does not exploit the special structure of the regularizer in Eq. (4.15); and thus it may not be efficient. Alternatively, in my dissertation study, I address the AFL problem though a carefully designed proximal gradient descent approach.

In this section, I evaluate the proposed structured sparse method with the AFL regularizer from the following respects. In synthetic studies, I first compare the computational efficiency between the DC-ADMM approach and the DC-Proximal approach. Evaluations are conducted in different scenarios, each of which demonstrates the relationship between the running time and some particular factors while keeping other factors unchanged. In real-world studies, I evaluate the AFL model on the ADNI data sets with two major objectives: (1) evaluating the prediction performance, and (2) identifying genetic risk factors—*i.e.*, AD-related SNPs. Comparisons

**Table 4.3:** SNPs appearing in multiple top lists selected by tree-structured group Lasso method. (Part 1)

| RS_ID | p-value | | | | Gene |
|---|---|---|---|---|---|
| | LEH | LHP | REH | RHP | |
| rs3745150 | 3.01e-04 | 3.86e-06 | 6.91e-07 | 1.16e-06 | PVRL2 |
| 19:45386467 | 2.08e-04 | 3.13e-07 | 4.10e-07 | 1.05e-07 | |
| rs12972156 | 2.08e-04 | 3.11e-07 | 4.08e-07 | 1.05e-07 | |
| rs12972970 | 2.08e-04 | 3.11e-07 | 4.07e-07 | 1.04e-07 | |
| rs283810 | 1.92e-03 | 1.27e-05 | 7.63e-06 | 5.39e-06 | |
| rs283811 | 3.43e-04 | 5.90e-08 | 2.59e-07 | 4.32e-08 | |
| rs283812 | 5.04e-04 | 1.04e-06 | 2.61e-06 | 1.17e-06 | |
| rs283814 | 5.06e-04 | 9.04e-07 | 3.43e-06 | 9.89e-07 | |
| rs283815 | 1.12e-04 | 1.12e-08 | 2.91e-08 | 2.35e-09 | |
| rs76692773 | 2.19e-04 | 2.17e-07 | 4.54e-07 | 6.80e-08 | TOMM40 |
| rs71352238 | 5.63e-04 | 5.89e-07 | 2.63e-06 | 5.01e-07 | |
| rs184017 | 9.03e-03 | 5.38e-04 | 4.82e-05 | 2.56e-05 | |
| rs2075649 | 2.24e-04 | 1.99e-07 | 4.20e-07 | 6.48e-08 | |
| rs2075650 | 5.26e-04 | 5.86e-07 | 2.47e-06 | 4.77e-07 | |
| rs157581 | 2.31e-03 | 9.96e-04 | 1.26e-05 | 3.81e-04 | |
| rs34095326 | 2.24e-04 | 2.06e-07 | 4.25e-07 | 6.74e-08 | |
| rs34404554 | 2.24e-04 | 2.08e-07 | 4.27e-07 | 6.79e-08 | |
| rs11556505 | 5.59e-04 | 6.25e-07 | 2.78e-06 | 5.36e-07 | |
| rs157582 | 1.91e-04 | 1.38e-07 | 1.30e-06 | 1.23e-07 | |
| 19:45406538 | 5.61e-03 | 5.24e-05 | 1.50e-04 | 2.03e-06 | |
| rs7259620 | 1.65e-03 | 1.54e-04 | 2.68e-04 | 3.61e-03 | APOE |
| rs405509 | 1.39e-02 | 3.43e-03 | 1.49e-03 | 1.37e-03 | |
| rs440446 | 4.64e-07 | 4.19e-11 | 1.06e-08 | 2.87e-11 | |
| rs769450 | 1.50e-07 | 2.68e-12 | 9.07e-10 | 1.75e-12 | |
| rs1081106 | 1.83e-04 | 1.04e-06 | 5.85e-05 | 3.71e-06 | none |
| rs445925 | 6.48e-07 | 3.77e-11 | 1.30e-08 | 2.99e-11 | APOC1 |
| rs10414043 | 6.40e-07 | 3.78e-11 | 1.29e-08 | 3.02e-11 | |
| rs7256200 | 3.66e-05 | 4.03e-08 | 3.07e-06 | 1.47e-07 | |
| rs584007 | 3.68e-05 | 4.04e-08 | 3.13e-06 | 1.48e-07 | |

**Table 4.4:** SNPs appearing in multiple top lists selected by tree-structured group Lasso method. (Part 2)

| RS_ID | p-value | | | | Gene |
|---|---|---|---|---|---|
| | LEH | LHP | REH | RHP | |
| rs390082 | 7.54e-05 | 1.74e-06 | 7.88e-06 | 2.79e-06 | APOC1 |
| 19:45417632 | 7.68e-05 | 1.81e-06 | 8.14e-06 | 2.86e-06 | |
| rs12691088 | 4.05e-05 | 1.05e-07 | 3.42e-06 | 2.04e-07 | |
| rs3826688 | 1.10e-06 | 2.30e-10 | 1.89e-08 | 7.56e-11 | |
| rs150966173 | 4.28e-06 | 1.05e-09 | 3.65e-08 | 3.54e-10 | |
| rs484195 | 1.14e-02 | 1.44e-03 | 2.45e-02 | 8.43e-04 | |
| 19:45421972 | 1.44e-06 | 9.39e-11 | 6.88e-09 | 2.32e-11 | |
| rs1064725 | 1.39e-06 | 9.44e-11 | 6.89e-09 | 2.21e-11 | |
| rs56131196 | 1.39e-06 | 9.43e-11 | 6.92e-09 | 2.21e-11 | |
| rs4420638 | 1.14e-02 | 1.45e-03 | 2.45e-02 | 8.52e-04 | |

have been conducted between the fused Lasso and AFL.

#### *4.6.4.1 Synthetic Study of AFL*

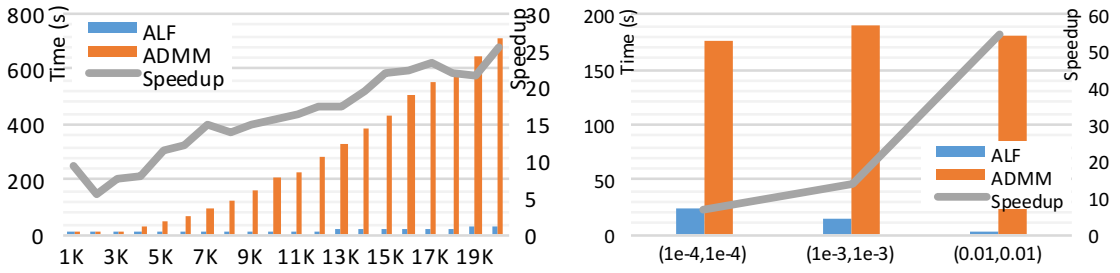
**Efficiency of AFL**

In the first series of experiments, I present some empirical studies on the efficiency of our proposed algorithm by comparing our method with the approach that adopts ADMM to solve the sub-problem at each DC iteration. The experiments are carried out on a collection of randomly generated data sets $\mathbf{A} \in \mathbb{R}^{n \times p}$ and outcomes $\mathbf{y} \in \mathbb{R}^{n \times 1}$. In addition, denote $\bar{\lambda} = \|\mathbf{A}^T\mathbf{y}\|_\infty$. I then conduct the evaluations in the following two scenarios:

1. **Varying the number of features $p$ with a fixed sample size and fixed regularization parameters $\lambda_1$ and $\lambda_2$.** I fix the number of samples $n = 500$ and vary the number of features $p$ from 1,000 to 20,000. I set the regularizers as $\lambda_1 = \lambda_2 = 10^{-3}\bar{\lambda}$.

2. **Varying regularization parameters $\lambda_1$ & $\lambda_2$ with a fixed sample/feature size**. I fix the $n = 500$ and $p = 10,000$. I choose the values of $(\lambda_1, \lambda_2)$ from the following set: $\{(10^{-4}\bar{\lambda}, 10^{-4}\bar{\lambda}), (10^{-3}\bar{\lambda}, 10^{-3}\bar{\lambda}), (0.01\bar{\lambda}, 0.01\bar{\lambda})\}$.

Figure 4.8 summarizes the running time (in seconds) and speedup of AFL (proximal algorithm) over ADMM in the above two scenarios. From these figures, it is easy to obtain the following observations: (1) The proposed algorithm is much more efficient than ADMM in both scenarios. (2) The speedup of AFL over ADMM increases as the feature size increases. This indicates that the proposed approach using DC programming and the proximal algorithm is capable of handling large-scale learning problems. (3) The speedup of AFL over ADMM increases as the regularized parameters become larger. In other words, the proposed method is expected to be superior over ADMM in real-world applications, as only a small number of features are relevant—*i.e.*, a relatively large regularized parameter value is preferred.




(a) Scenario 1 (b) Scenario 2

**Figure 4.8:** Comparison of running times and speedups of DC–Proximal (AFL) over DC–ADMM.

### Comparison of AFL and Fused Lasso

In this section, I compare the AFL model with the fused Lasso. Recall that the AFL is designed to encourage the smoothness of adjacent coefficients whose absolute values are close or even identical. Thus if the adjacent features exhibit different signs in the

model, the AFL approach is expected to be more effective than the fused Lasso in general.

I generate the synthetic data via a linear model $\mathbf{y} = \mathbf{A}\bar{\mathbf{x}} + \epsilon$, where the design matrix $\mathbf{A} \in \mathbb{R}^{500\times5000}$ and the noise term $\epsilon \in \mathbb{R}^n$ are randomly generated from normal distributions. The ground truth $\bar{\mathbf{x}} \in \mathbb{R}^n$ contains 10% of the signals, which are evenly partitioned into 5 groups. Specifically, within each group, I first continuously assign the same value for all the signals; and then, I randomly pick $\{0\%, 1\%, 2\%, 5\%, 10\%\}$ of the signals and change their signs to the opposite. Regularization parameters $\lambda_1$ and $\lambda_2$ are chosen from the interval $[10^{-4}\bar{\lambda}, 0.9\bar{\lambda}]$ using five-fold cross-validation for both the AFL and the fused Lasso. I then evaluate the models on a 100 i.i.d. samples testing set. The SLEP package [Liu *et al.* (2009b, 2010)] is adopted to solve the fused Lasso problem. I report the averaged predictive performance of 10 replications in Table 4.5.

**Table 4.5:** Averaged prediction performance of the AFL method and the fused Lasso on synthetic data (standard deviation is shown in the bracket). FL refers to the fused Lasso. MSE refers to the mean squared error. Corr_X is the Pearson correlation between the model $\mathbf{x}$ and the ground truth $\bar{\mathbf{x}}$.

| Neg% | Method | MSE_Y | MSE_X | Corr_X |
|---|---|---|---|---|
| 0% | AFL | 0.0001 (0.00) | 0.0000 (0.00) | 1.00 (0.00) |
| | FL | 0.0003 (0.00) | 0.0000 (0.00) | 1.00 (0.00) |
| 1% | AFL | 0.0157 (0.02) | 0.0000 (0.00) | 1.00 (0.00) |
| | FL | 0.0051 (0.00) | 0.0000 (0.00) | 1.00 (0.00) |
| 2% | AFL | 0.0179 (0.01) | 0.0000 (0.00) | 1.00 (0.00) |
| | FL | 0.0227 (0.01) | 0.0000 (0.00) | 1.00 (0.00) |
| 5% | AFL | 15.16 (11.09) | 0.0029 (0.00) | 0.98 (0.01) |
| | FL | 51.75 (23.55) | 0.0103 (0.00) | 0.92 (0.04) |
| 10% | AFL | 86.32 (28.21) | 0.0200 (0.00) | 0.81 (0.03) |
| | FL | 125.98 (19.85) | 0.0242 (0.00) | 0.78 (0.01) |

It can be observed from Table 4.5 that the AFL approach provides better predictive performance than the fused Lasso in most cases. If the ground truth $\bar{\mathbf{x}}$ does not contain too many opposite adjacent signals, both AFL and the fused Lasso can accurately recover the original signals. However, when the number of opposite signals increases, AFL outperforms the fused Lasso significantly. The reason is that, with the AFL penalty, the model tends to select those highly similar adjacent features even if their signs are different. Therefore, the AFL approach is more robust than the fused Lasso in such cases.

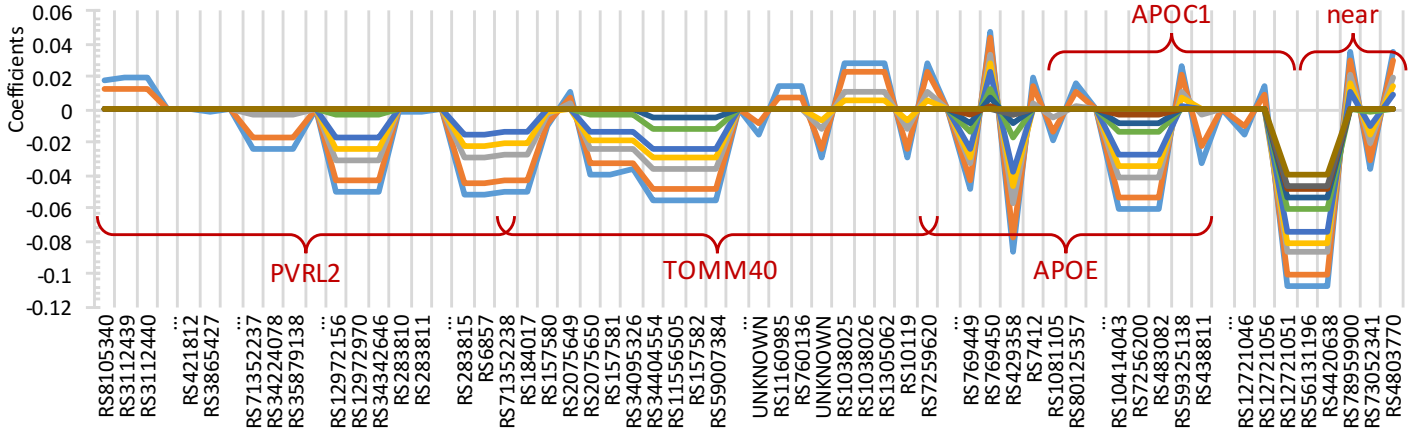
#### *4.6.4.2 ADNI Imaging Genetics Study*

In the following section, I evaluate the AFL model on the ADNI whole genome sequence data. Particularly, I investigate imaging genetics associations between imaging phenotypes and SNPs (within the 19th chromosome) using the regression model with the AFL penalty. The baseline entorhinal cortex (EC) and hippocampal (HIPP) volumes are chosen to be the responses, as these are two major brain regions affected by the Alzheimer's disease.

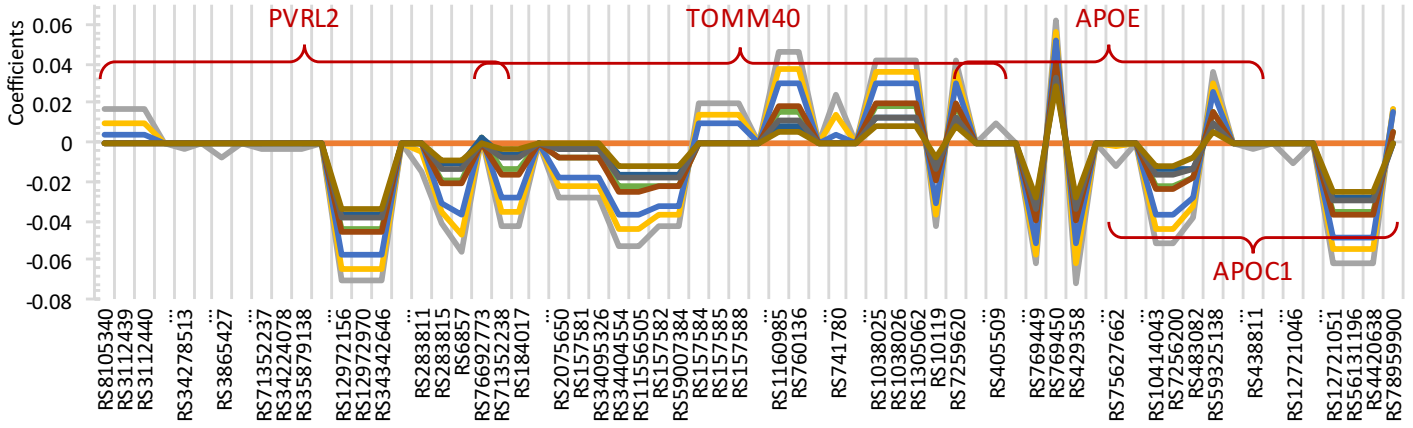
**Detecting Risk Genetic Factors using AFL**

Inspired by the idea of interaction testing introduced in Bien *et al.* (2015), I conduct a study on detecting AD risk genetic factors with the AFL model. Specifically, on Chromosome 19, I first calculate the Pearson correlation between each coded SNP and the response imaging phenotype vector. Then, I plug the correlation coefficients vector into our model (4.11). To identify the most association SNPs, I vary the regularization parameters and record each model.

Figure 4.9 shows the study results of using EC and HIPP as responses. In the experiment, we can observe that the AFL model can successfully capture AD risk




(a) Entorhinal




(b) Hippocampus

**Figure 4.9:** Regression coefficients learned by each AFL model. Each color in the graph represents a learned model based on a pair of regularizers $(\lambda_1, \lambda_2)$. SNPs (named by RS_IDs) are presented in their order on Chr.19. "..." indicates the gaps between SNPs. AD risk genes are marked in red.

genes including PVRL2 [Logue *et al.* (2011)], TOMM40 [Maruszak *et al.* (2012); Guerreiro and Hardy (2012); Lyall *et al.* (2014)], APOE [Logue *et al.* (2011); Maruszak *et al.* (2012); Lyall *et al.* (2014); Tycko *et al.* (2004)] and APOC1 [Zhou *et al.* (2014); Tycko *et al.* (2004)]. Moreover, the AFL is capable of performing automatic feature grouping even when the signs are different, e.g., rs769449, rs769450 and rs429358 in APOE exhibit high similarity in absolute values. However, the fused Lasso fails to correctly group SNPs like rs769450 since their signals are different. In Table 4.6, I further present some statistical scores of SNPs selected by the AFL model, including p-value [1] (P) and odds ratio (OR) association score. It can be observed that most of the selected SNPs achieve high statistical significance.

### *4.6.5 sgLasso_gGraph for ADNI Imaging Genetics*

In this section, I evaluate the proposed sgLasso_gGraph on ADNI imaging genetics data set. More specifically, as mentioned in Section 4.4, I utilize two gene networks based on a set of 10 pre-selected AD candidate genes on the 19th chromosome. The gene selection is according to AlzGene, and the candidate gene including LDLR, GAPDHS, BCAM, PVRL2, TOMM40, APOE, APOC1, APOC4, EXOC3L2, and CD33. To obtain gene networks, I utilize GeneMANIA to explore existing network data. A potential gene-gene relationship includes protein or genetic interactions, pathways, co-expression, co-localization or protein domain similarity. In the later experiment, I also extend the size of candidate gene set to 20. The additional genes are introduced according to the gene ontology weights from the biological process. As a consequence, the connections of gene networks are presented in Figures 4.4–4.5, and detailed statistics of those genes are available in Appendix E.

[1] Those p-values are obtained from Pearson correlation analysis between SNPs and the selected imaging phenotype.

**Table 4.6:** Statistical scores of selected SNPs on Chromosome 19. P-EC refers to the p-value associated with the EC task. P-HIPP refers to the p-value associated with the HIPP task. OR refers to the odds ratio associated with MCI&AD.

| RS_ID | Gene | P-EC | P-HIPP | OR |
|---|---|---|---|---|
| rs12972156 | PVRL2 | 1.03e-04 | 1.23e-05 | 1.947 |
| rs12972970 | PVRL2 | 1.24e-04 | 9.98e-06 | 1.984 |
| rs34342646 | PVRL2 | 1.18e-04 | 9.51e-06 | 1.809 |
| rs283815 | PVRL2 | 1.98e-04 | 1.17e-03 | 1.436 |
| rs6857 | PVRL2 | 8.07e-06 | 2.05e-06 | 1.914 |
| rs76692773 | PVRL2∼TOMM40 | 3.86e-01 | 2.64e-01 | 0.912 |
| rs71352238 | PVRL2∼TOMM40 | 9.20e-05 | 1.32e-05 | 1.767 |
| rs184017 | TOMM40 | 2.72e-05 | 8.31e-04 | 1.414 |
| rs2075650 | TOMM40 | 5.33e-04 | 3.15e-04 | 1.791 |
| rs157581 | TOMM40 | 5.43e-05 | 1.39e-03 | 1.436 |
| rs34095326 | TOMM40 | 4.14e-02 | 6.25e-02 | 1.511 |
| rs34404554 | TOMM40 | 1.59e-04 | 4.42e-05 | 1.842 |
| rs11556505 | TOMM40 | 1.60e-04 | 4.23e-05 | 1.857 |
| rs157582 | TOMM40 | 8.06e-05 | 1.96e-03 | 1.435 |
| rs59007384 | TOMM40 | 5.20e-05 | 5.13e-04 | 1.541 |
| rs769449 | APOE | 1.54e-05 | 3.30e-06 | 2.646 |
| rs769450 | APOE | 9.99e-03 | 2.87e-03 | 0.897 |
| rs429358 | APOE | 2.13e-08 | 2.50e-07 | 2.409 |
| rs10414043 | APOE∼APOC1 | 1.49e-05 | 3.17e-05 | 2.447 |
| rs7256200 | APOE∼APOC1 | 1.96e-05 | 6.68e-05 | 2.447 |
| rs483082 | APOE∼APOC1 | 1.30e-04 | 1.55e-03 | 1.690 |
| rs12721051 | APOC2 | 1.73e-07 | 8.62e-06 | 1.914 |
| rs56131196 | APOC3 | 3.44e-08 | 5.11e-05 | 1.739 |
| rs4420638 | APOC4 | 3.40e-08 | 6.77e-05 | 1.712 |
| rs78959900 | APOC1 | 3.28e-02 | 1.27e-01 | 0.899 |
| rs73052341 | APOC1 | 4.65e-05 | 3.97e-05 | 1.978 |

**Candidate Genes within Chromosome 19**

In this experiment, I use the ADNI WGS SNPs data of the ten AD candidate genes on Chromosome 19. As a consequence, the experimental data set contains 1,381 subjects with 504 SNPs. I use four outcomes in this study, including volumes of the left entorhinal cortex (LEH), left hippocampus (LHP), right entorhinal cortex (REH), and right hippocampus (RHP). To evaluate the predictive performance of the proposed method, I compare the sgLasso_gGraph method with the Lasso model, with a fix the parameter that controls the sparsity of SNPs-level. The average predictive performance of 10 replications together with five-fold cross-validation are summarized in Table 4.7.

Note that the original outcomes are not well aligned and thus it may not be appropriate to used them as learning target directly. To this end, I use the following schema to prepare the outcomes according to their physical meaning. Specifically, for each response, I first take the cube root and then center them around zero; similarly hereinafter.

**Candidate Genes among Chromosome 19 with Other Genes**

Similar to the previous section, but this experimental data set includes 10 more genes (*i.e.* 20 genes in total). The additional genes are selected via GeneMANIA, according to the gene ontology weights from the biological process. The corresponding network connections are shown in Figure 4.5. As a consequence, the experimental data set contains 1,381 subjects with 1,364 SNPs. I use the same setting as the previous experiment; averaged experimental results of 10 replications together with five-fold cross-validation are summarized in Table 4.8.

From Table 4.7 and Table 4.8, I have the following favorable observations: (1) For prediction tasks LEH and REH, the proposed two-level structured sparse model,

*i.e.* sgLasso_gGraph, outperforms the basic Lasso models significantly in terms of MSE. (2) For the other two brain regions LHP and RHP, the predictive abilities of Lasso model and sgLasso_gGraph model are similar. However, when more gene-level network information is available, the proposed sgLasso_gGraph method is expected to be superior over Lasso (see Table 4.8). Those experimental results imply that it is beneficial to incorporate gene-level networks knowledge during model fitting.

### *4.6.6 CNN for ADNI Imaging Genetics*

Besides the aforementioned structured sparse methods, in my dissertation, I also consider adopting deep learning techniques—specifically, the convolutional neural networks for ADNI imaging genetics. As shown in Figure 4.2, I adopt the following CNN architecture:

$$INPUT \rightarrow CONV \rightarrow ReLU \rightarrow POOL \rightarrow FC \rightarrow (+Dropout) \rightarrow ReLU \rightarrow FC.$$

More specifically, the input of the proposed CNN model is a one-dimensional vector of SNP sequence. The input layer is followed by a convolutional layer of window size 3, stride size 1, and 5 filters. I use the ReLU as the activation function for the CONV layer. Then the max-pooling operation of window size 2 and stride size 2, is performed on each filter of the CONV layer. After flattening, I adopt two fully connected layers of sizes 256 and 1 to transform multiple neurons to a single one to represent the final output (*i.e.* an imaging phenotype). Again, the ReLU is used as the activation function for the first FC layer. In addition, a dropout probability of 0.5 is adopted at the first FC layer to alleviate overfitting during the training procedure. For experiments, I use the same data sets presented in Section 4.6.5; preliminary experimental results of 10 replications together with five-fold cross-validation are summarized in Table 4.7 and Table 4.8.

It can be observed from Tables 4.7 and 4.8 that, the CNN approach produce the

best overall prediction performance in terms of MSE among all tasks. This implies that adopting convolutional neural networks is promising in imaging genetics studies, especially for predicting image-based biomarkers based on genomic data.

However, it is worth mentioning that, a potential problem of using CNN in imaging genetics is that, it may not be easy to scale the network, as the input data is a one-dimensional SNPs sequence. More specifically, such a problem is major caused by the fully connected layers in the neural networks. Suppose we are plugging in a sequence of tens of thousands of SNPs in length, the number of parameters within two fully connected layers will be huge. This could lead to two direct drawbacks: (1) it requires large memory and increases computational costs; and (2) it is easier to get overfit if we only have thousand of training samples, *i.e.*, the predictive performance on the testing set will be poor.

**Table 4.7:** Comparison between Lasso, sgLasso_gGraph and CNN approaches in terms of MSE on candidate genes within Chromosome 19. Standard deviation is shown in the bracket.

| Response | Lasso | sgLasso_gGraph | CNN |
|---|---|---|---|
| LEH | 1.2812 (0.1118) | 1.2531 (0.1057) | 1.2498 (0.1072) |
| LHP | 0.8766 (0.0605) | 0.8732 (0.0591) | 0.8699 (0.0584) |
| REH | 1.1831 (0.1046) | 1.1598 (0.1056) | 1.1216 (0.1104) |
| RHP | 0.9102 (0.0943) | 0.9116 (0.0946) | 0.8958 (0.1521) |

**Table 4.8:** Comparison between Lasso, sgLasso_gGraph and CNN approaches in terms of MSE on extended gene networks based on Chromosome 19 candidate genes. Standard deviation is shown in the bracket.

| Response | Lasso | sgLasso_gGraph | CNN |
|---|---|---|---|
| LEH | 1.3337 (0.1163) | 1.2568 (0.1058) | 1.2338 (0.0949) |
| LHP | 0.9054 (0.0654) | 0.8686 (0.0646) | 0.8594 (0.0549) |
| REH | 1.1911 (0.1073) | 1.1387 (0.1020) | 1.0908 (0.0829) |
| RHP | 0.9509 (0.0815) | 0.9164 (0.0855) | 0.9039 (0.1491) |

### 4.6.7 Comparison between Different Structured Sparse Methods

In the last series of experiments, I compare a suite of commonly used structured sparse methods, including Lasso, the fused Lasso, and sparse group Lasso, with three approaches proposed in this dissertation, *i.e.*, the absolute fused Lasso, the two-level structured sparse model (*a.k.a.*, sgLasso_gGraph), and the CNN method. For a fair comparison, both of the experiments are conducted based on the data set introduced in Section 4.4. More specifically, the experimental data set contains the SNPs of 10 pre-selected AD-risk gene on the 19th chromosome. For SGL and sgLasso_gGraph, SNPs in the same gene fall into a group in the model. Again, four neuroimaging phenotypes including volumes of the left entorhinal cortex (LEH), left hippocampus (LHP), right entorhinal cortex (REH), and right hippocampus (RHP) are used as responses in this study.

#### 4.6.7.1 Rregression Tasks

In this experiment, I investigate the predictive performance of different structured sparse methods on ADNI imaging genetics data. Here I adopt the five-fold cross-validation for each learning method. The predictive performance in terms of MSE of 10 replications are shown in Figure 4.10 through boxplot. In the figure, each color represents a modeling method and the first three letters in the label of x-axis indicate a learning task.

From Figure 4.10, I have the following observations: (1) For most of the cases, the proposed novel structured sparse methods outperform traditional models. (2) Although different models produce similar predictive performance, such as in LHP tasks, the proposed novel structured sparse methods are still interesting, as they have incorporated different biological prior knowledge into the models. As a consequence,

such models have better interpretability than the traditional ones. (3) Although the CNN approach produces the state-of-the-art overall performance in most of the tasks, it is not stable (more outliers in the boxplot). This is potentially caused by the limited number of training examples. To sum up, the above experimental results indicate that, it is beneficial to address real-world imaging genetics problems by incorporating different biological prior knowledge through carefully-designed sparse-inducing regularizers.

#### *4.6.7.2 Model Selection Tasks*

Recall that in imaging genetics studies, identifying disorder-related genetic risk SNPs is one of the major tasks. In this section, I compare the model selection results of different structured sparse methods through stability selection. More specifically, for each outcome, I perform 100 simulations. In each simulation, I first randomly subsample half of the samples and then perform a modeling method 100 times with different regularization parameters (or pairs of parameters). The model selection results are reported in Figure 4.11. In the figure, the top 50 selected SNPs are marked for each method, each color refers to a structured sparse method, and the x-axis indicates the SNPs' location in the data. The green bars are the negative of logarithmic of p-values of the corresponding SNPs.

From Figure 4.11, we have the following observations: (1) SNPs selected by Lasso and sparse group Lasso models are spread over a large region in the feature sets. However, most SNPs selected by the fused Lasso, absolute fused Lasso, and sgLasso_gGraph models are clustered in a few small regions. (2) In comparing between the FL and AFL, the later one produces better local smoothness. (3) SNPs' groups obtained by sgLasso_gGraph are different from FL or AFL, and those selected SNPs are not necessary to come with small p-values (see the bottom two sub-figures

in Figure 4.11). It is worth mentioning that such a scenario is very interesting, as it may be caused by gene-level interactions. In addition, it is well known that, in genetics, the aggregate effects of multiple SNPs are more significant than individual effects. Therefore, the above observations demonstrate that the proposed structured sparse methods are able to identify groups of causal SNPs that related to a disorder.

## 4.7 Summary

In this chapter, I introduce several research works on the ADNI imaging genetics data sets. A key idea in those works is to adopting different structured sparse methods for model construction. Due to their capability of incorporating various prior knowledge, sparse models are very effective in identifying the predictors that exhibit the strongest effects on the imaging phenotypes. Specifically, I first adopt Lasso and EDPP screening rules to effectively AD-risk SNPs. Next, I utilize tree-structured group Lasso to incorporate LD information into the model and MLFre screening rules for fast computation. Moreover, I propose a absolute fused Lasso, which can be considered as a robust extension of the fused Lasso, for ADNI imaging genetics study. The AFL takes advantages of the SNP spatial structure and is robust to the choice of reference alleles during genome-type coding. In addition, I further develop a two-level structured sparse model, which is capable of utilizing gene-level networks on SNP-level data study. This approach can also be considered as a sparse group Lasso model with a group-level graph structure. In the last part, I propose to adopt a convolutional neural network with dropout technique for accurate predicting imaging phenotypes based on SNPs data. Experimental results show that structured sparse methods are powerful tools in facilitating the research of imaging genetics.

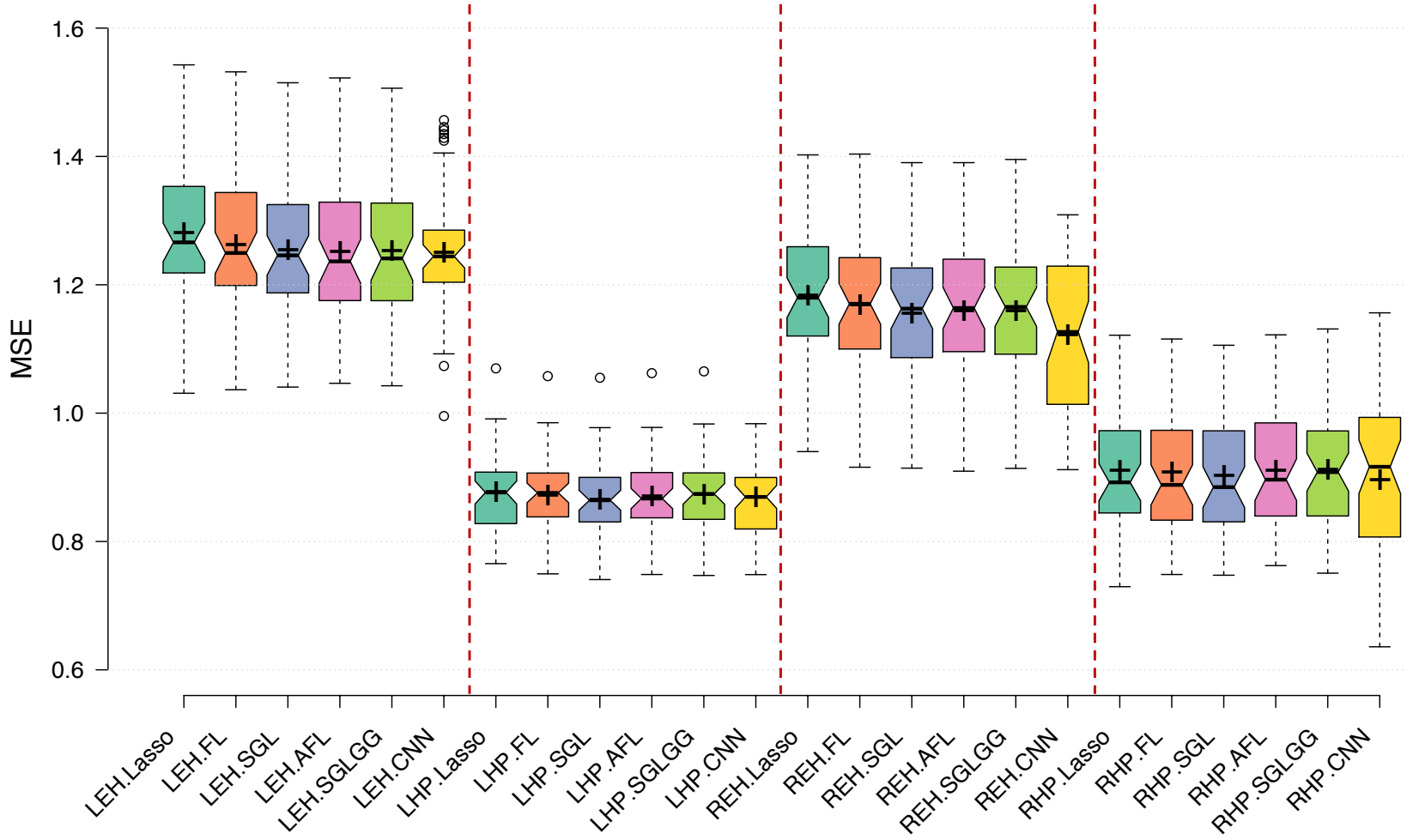



**Figure 4.10:** Comparison of regression error in terms of MSE between different structured sparse models on candidate AD-risk genes on Chr19.

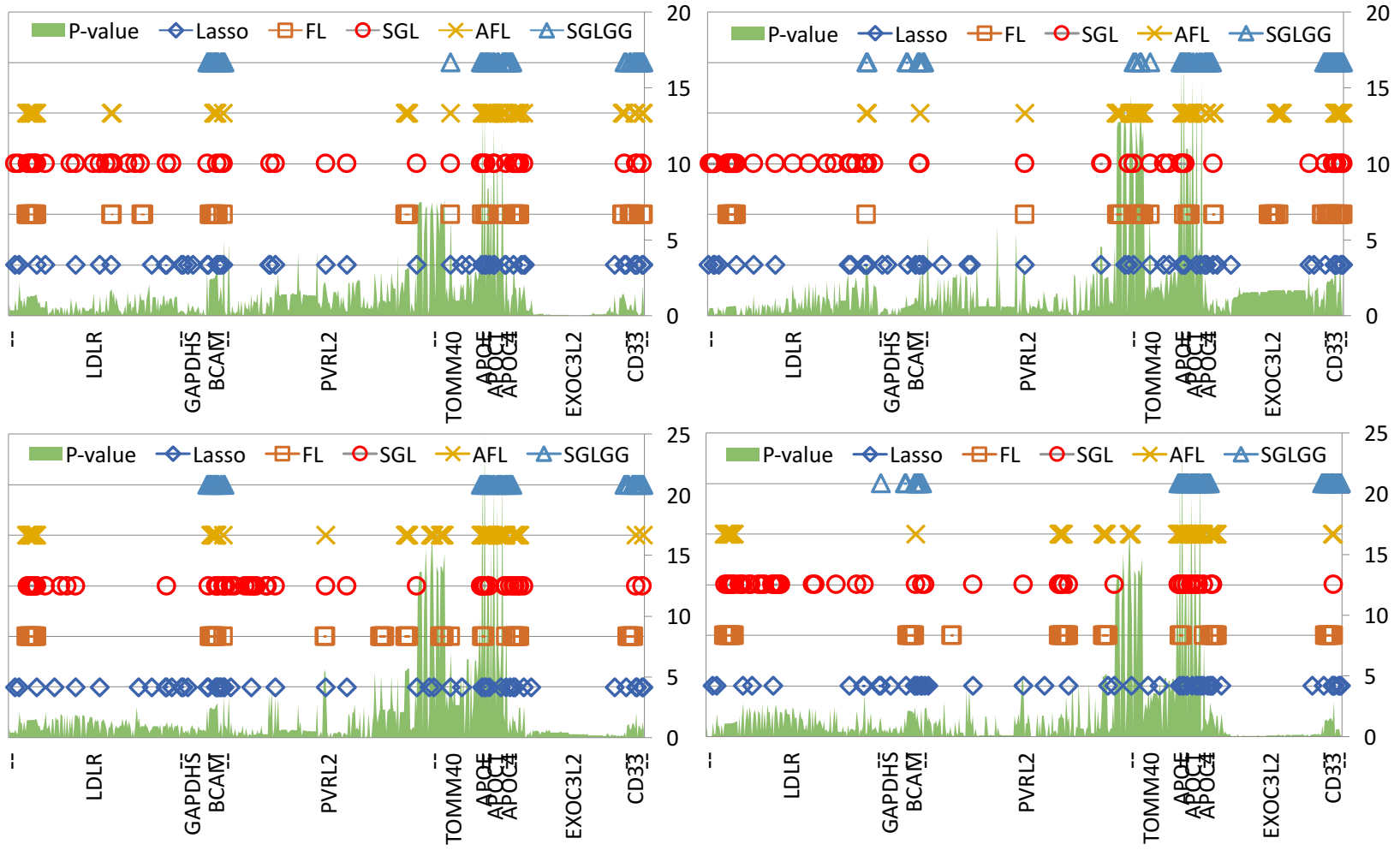



**Figure 4.11:** Comparison of stability selection results (top 50 SNPs) between different structured sparse models on candidate genes on Chr19. UL-LEH. BL-LHP. UR-REH. BR-RHP.

# Chapter 5

# CONCLUSION AND FUTURE WORK

In this chapter, I summarize the major contributions of my dissertation research. In addition, I discuss some possible future work of applying structured sparse methods in imaging genetics.

## 5.1 Summary of Contributions

In my dissertation study, I carry on my research on real-world imaging genetics applications with particular focuses on the following two directions: (1) building effective predictive models between imaging phenotypes and molecular genomic data, and (2) identifying major disorder-related genetic risk factors—*i.e.*, SNPs, through imaging phenotypes. To address those two objectives, I consider a suite of structured sparse methods for imaging genetics studies. There are several benefits of adopting such structured sparse methods. First and foremost, sparse methods can perform simultaneously regression (model fitting) as well as variable selection. Secondly, introducing sparsity is a good way to alleviate overfitting during model learning, especially in real-world imaging genetics studies with the curse of high-dimensionality on both imaging data as well genomic data. Furthermore, with carefully-designed sparse-inducing penalties, different biological priors can be incorporated into sparse models. This provides the model better interoperability.

More specifically, in my first real-world application—Allen brain imaging – gene expression study, I focus on predicting (annotating) gene expression statuses based on raw image data sets. To generalize high-level image representations, I adopt an

advanced sparse coding method over the SIFT image descriptors. For the multi-class annotation subtask, I address this problem by utilizing a multi-task learning approach which taking advantages of the $\ell_{2,1}$-norm based group sparse structure, as the multiple classes are potentially inter-connected. In addition, I proposed a novel label structure-based two-stage multi-label learning framework. This work utilizes the hierarchical structure of the brain ontology. Experimental results show my proposed approaches outperform the state-of-the-art methods in most of the cases.

In the later ADNI imaging genetics research, I focus on predicting disorder-related imaging phenotypes based on SNPs data, as well as identifying disease-related genetic risk factors (SNPs). To address the above two problems, I consider a suite of structured sparse methods, including Lasso, tree-structured group Lasso, absolute fused Lasso, a two-level structured sparse model (*i.e.* sparse group Lasso with group-level graph structure), and convolutional neural networks. Specifically, Lasso and EDPP screening rules are used as the basic model for ADNI imaging genetics. It is worth mentioning that, although Lasso cannot include further biological priors, it is the most efficient model and allows us to investigate the genome-wide associations over the entire genome. In a later study, I adopt tree-structured group Lasso together with MLFre screening rule for ADNI imaging genetics studies. The TGL model incorporates LD information over SNPs. In the sequel, I propose the absolute fused Lasso as an extension of the fused Lasso which takes advantages of SNP spatial structure. Although AFL is a non-convex model, it is more robust to the choice of reference alleles during genome sequence data processing. Moreover, I propose a sgLasso_gGraph model that incorporates gene-level network data as graphs into SNP-level model construction. This is beneficial since there are considerable existing studies on gene-/protein-level interactions. In the last part of my dissertation work, I explore convolutional neural networks in imaging genetics. Although it is

not capable of identifying genetic risk factors for a disorder, CNN can often provide the state-of-the-art predictive performance of imaging phenotypes. Experiments have been conducted based on the ADNI WGS SNPs data and T1 MRI data. Preliminary results demonstrate the efficiency and the effectiveness of the proposed structured sparse methods.

## 5.2 Future Directions

There are several future works for applying structured sparse methods in imaging genetics studies. First of all, for the Allen study, a promising direction is to adopt deep learning techniques in the annotation tasks. Some research topics—for example, "shall we learn four stages simultaneously or independently?", "what are the appropriate learning targets in deep learning? (shall we use the entire brain ontology as outcomes?)"—are very interesting as well as valuable. On the other hand, for ADNI imaging genetics study, it is still very hard to incorporate complex biological prior knowledge into the model. Difficulties are mainly due to the following three aspects: (1) no unified database or resource would provide a complete biological knowledge base, (2) different data sets are not well aligned between multiple imaging genetics global consortiums, and (3) existing models are not as effective as expected in the high-dimensional scenario. Last, and most importantly in future imaging genetics, it is of urgent importance to collect research data world-widely (*i.e.* collect more experimental examples) and build a unified database for research purpose.

Allen Institute for Brain Science, "Allen Developing Mouse Brain Atlas", Tech. rep., URL `http://developingmouse.brain-map.org` (2013).

Ando, R. K. and T. Zhang, "A framework for learning predictive structures from multiple tasks and unlabeled data", The Journal of Machine Learning Research **6**, 1817–1853 (2005).

Argyriou, A., T. Evgeniou and M. Pontil, "Convex multi-task feature learning", Machine Learning **73**, 3, 243–272 (2008).

Ashburner, M., C. Ball, J. Blake, D. Botstein and H. Butler, "Gene ontology: tool for the unification of biology. the gene ontology consortium", Nature Genetics **25**, 25–29 (2000).

Bach, F., R. Jenatton, J. Mairal, G. Obozinski *et al.*, "Structured sparsity through convex optimization", Statistical Science **27**, 4, 450–468 (2012).

Bakker, B. and T. Heskes, "Task clustering and gating for bayesian multitask learning", The Journal of Machine Learning Research **4**, 83–99 (2003).

Batmanghelich, N. K., A. V. Dalca, M. R. Sabuncu and P. Golland, "Joint modeling of imaging and genetics", in "Information Processing in Medical Imaging", pp. 766–777 (Springer, 2013).

Baxter, J., "A bayesian/information theoretic model of learning to learn via multiple task sampling", Machine Learning **28**, 1, 7–39 (1997).

Baxter, J., "A model of inductive bias learning", J. Artif. Intell. Res.(JAIR) **12**, 149-198, 3 (2000).

Beck, A. and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems", SIAM Journal on Imaging Sciences **2**, 1, 183–202 (2009).

Ben-David, S. and R. Schuller, "Exploiting task relatedness for multiple task learning", in "Learning Theory and Kernel Machines", pp. 567–580 (Springer, 2003).

Bertram, L., M. B. McQueen, K. Mullin, D. Blacker and R. E. Tanzi, "Systematic meta-analyses of Alzheimer disease genetic association studies: the alzgene database", Nature genetics **39**, 1, 17–23 (2007).

Bettens, K., K. Sleegers and C. Van Broeckhoven, "Genetic insights in Alzheimer's disease", The Lancet Neurology **12**, 1, 92–104 (2013).

Bi, W. and J. T. Kwok, "Multi-label classification on tree-and dag-structured hierarchies", in "Proceedings of the 28th ICML", pp. 17–24 (2011).

Bien, J., N. Simon, R. Tibshirani *et al.*, "Convex hierarchical testing of interactions", The Annals of Applied Statistics **9**, 1, 27–42 (2015).

Bis, J. *et al.*, "Common variants at 12q14 and 12q24 are associated with hippocampal volume", Nature genetics **44**, 5, 545–551 (2012).

Bondell, H. D. and B. J. Reich, "Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with OSCAR", Biometrics **64**, 1, 115–123, URL `http://dx.doi.org/10.1111/j.1541-0420.2007.00843.x` (2008).

Boureau, Y.-L., J. Ponce and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition", in "Proceedings of the 27th ICML", pp. 111–118 (2010).

Boutell, M. R., J. Luo, X. Shen and C. M. Brown, "Learning multi-label scene classification", Pattern recognition **37**, 9, 1757–1771 (2004).

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

Bruckstein, A., D. Donoho and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images", SIAM Review **51**, 34–81 (2009).

Candès, E. J. *et al.*, "Compressive sampling", Proceedings of the international congress of mathematicians **3**, 1433–1452 (2006).

Caruana, R., "Multitask learning", Machine learning **28**, 1, 41–75 (1997).

Chang, C.-C. and C.-J. Lin, "LIBSVM: A library for support vector machines", ACM Transactions on Intelligent Systems and Technology **2**, 27:1–27:27, software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm` (2011).

Chen, S. S., D. L. Donoho and M. A. Saunders, "Atomic decomposition by basis pursuit", SIAM J. Sci. Comput. **20**, 1, 33–61 (1998).

Chen, S. S., D. L. Donoho and M. A. Saunders, "Atomic decomposition by basis pursuit", SIAM Review **43**, 129–159 (2001).

Cirulli, E. T. and D. B. Goldstein, "Uncovering the roles of rare variants in common disease through whole-genome sequencing", Nature Reviews Genetics **11**, 6, 415–425 (2010).

Cornelis, M., L. Qi, C. Zhang, P. Kraft, J. Manson, T. Cai, D. Hunter and F. Hu, "Joint effects of common genetic variants on the risk for type 2 diabetes in US men and women of European ancestry", Annals of internal medicine **150**, 541–550 (2009).

Csurka, G., C. Dance, L. Fan, J. Willamowski and C. Bray, "Visual categorization with bags of keypoints", in "Workshop on statistical learning in computer vision, ECCV", vol. 1, pp. 1–2 (Prague, 2004).

Dembszynski, K., W. Waegeman, W. Cheng and E. Hüllermeier, "On label dependence in multilabel classification", in "LastCFP: ICML Workshop on Learning from Multi-label data", (Ghent University, KERMIT, Department of Applied Mathematics, Biometrics and Process Control, 2010).

Dinu, I., S. Mahasirimongkol, Q. Liu, H. Yanai, N. Eldin, E. Kreiter, X. Wu, S. Jabbari, K. Tokunaga and Y. Yasui, "SNP-SNP interactions discovered by logic regression explain Crohns disease genetics", PLoS One **7**, e43035 (2012).

Donoho, D. L. and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via $l1$ minimization", Proceedings of the National Academy of Sciences **100**, 5, 2197–2202 (2003).

El Ghaoui, L., V. Viallon and T. Rabbani, "Safe feature elimination in sparse supervised learning", Pacific Journal of Optimization **8**, 667–698 (2012).

Evgeniou, A. and M. Pontil, "Multi-task feature learning", Advances in neural information processing systems **19**, 41 (2007).

Evgeniou, T., C. A. Micchelli and M. Pontil, "Learning multiple tasks with kernel methods", in "Journal of Machine Learning Research", pp. 615–637 (2005).

Evgeniou, T. and M. Pontil, "Regularized multi–task learning", in "Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 109–117 (ACM, 2004).

Friedman, J., T. Hastie, H. Höfling, R. Tibshirani *et al.*, "Pathwise coordinate optimization", The Annals of Applied Statistics **1**, 2, 302–332 (2007).

Friedman, J., T. Hastie and R. Tibshirani, "A note on the group lasso and a sparse group lasso", arXiv preprint arXiv:1001.0736 (2010).

Fürnkranz, J., E. Hüllermeier, E. L. Mencía and K. Brinker, "Multilabel classification via calibrated label ranking", Machine learning **73**, 2, 133–153 (2008).

Golub, T. R., D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri and C. D. Bloomfield, "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring", Science **286**, 5439, 531–537 (1999).

Gong, P., C. Zhang, Z. Lu, J. Huang and J. Ye, "A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems", in "The 30th International Conference on Machine Learning (ICML)", pp. 37–45 (2013).

Guerreiro, R. J. and J. Hardy, "TOMM40 association with Alzheimer disease: tales of APOE and linkage disequilibrium", Archives of neurology **69**, 10, 1243–1244 (2012).

Guyon, I., J. Weston, S. Barnhill and V. Vapnik, "Gene selection for cancer classification using support vector machines", Machine Learning **46**, 1-3, 389–422, URL `citeseer.ist.psu.edu/guyon02gene.html` (2002).

Hariri, A. R., E. M. Drabant and D. R. Weinberger, "Imaging genetics: perspectives from studies of genetically driven variation in serotonin function and corticolimbic affective processing", Biological psychiatry **59**, 10, 888–897 (2006).

Harris, M. *et al.*, "The gene ontology database and informatics resource", Nucleic Acids Research **32**, 258–261 (2004).

Hastie, T., R. Tibshirani and M. Wainwright, *Statistical learning with sparsity: the lasso and generalizations* (CRC Press, 2015).

Hebert, L. E., J. Weuve, P. A. Scherr and D. A. Evans, "Alzheimer disease in the united states (2010–2050) estimated using the 2010 census", Neurology **80**, 19, 1778–1783 (2013).

Hochreiter, S., "Untersuchungen zu dynamischen neuronalen netzen", Diploma, Technische Universität München p. 91 (1991).

Huang, Y. and L. Mucke, "Alzheimer mechanisms and therapeutic strategies", Cell **148**, 6, 1204–1222 (2012).

Ioannidis, J. P., P. Boffetta, J. Little, T. R. OBrien, A. G. Uitterlinden, P. Vineis, D. J. Balding, A. Chokkalingam, S. M. Dolan, W. D. Flanders *et al.*, "Assessment of cumulative evidence on genetic associations: interim guidelines", International journal of epidemiology **37**, 1, 120–132 (2008).

James, G., D. Witten, T. Hastie and R. Tibshirani, *An introduction to statistical learning*, vol. 112 (Springer, 2013).

Juottonen, K., M. Lehtovirta, S. Helisalmi, P. Riekkinen Sr and H. Soininen, "Major decrease in the volume of the entorhinal cortex in patients with Alzheimers disease carrying the apolipoprotein e $\varepsilon 4$ allele", Journal of Neurology, Neurosurgery & Psychiatry **65**, 3, 322–327 (1998).

Kim, S. and E. P. Xing, "Tree-guided group lasso for multi-task regression with structured sparsity", in "Proceedings of the 27th International Conference on Machine Learning (ICML-10)", pp. 543–550 (2010).

Kowalski, M., "Spare regression using mixed norms", Applied and Computational Harmonic Analysis **27**, 303–324 (2009).

Lambert, J.-C., C. A. Ibrahim-Verbaas, D. Harold, A. C. Naj, R. Sims, C. Bellenguez, G. Jun, A. L. DeStefano, J. C. Bis, G. W. Beecham *et al.*, "Meta-analysis of 74,046 individuals identifies 11 new susceptibility loci for Alzheimer's disease", Nature genetics **45**, 12, 1452–1458 (2013).

LeCun, Y., Y. Bengio and G. Hinton, "Deep learning", Nature **521**, 7553, 436–444 (2015).

Li, C. and H. Li, "Network-constrained regularization and variable selection for analysis of genomic data", Bioinformatics **24**, 1175–1182 (2008).

Li, Q., T. Yang, L. Zhan, D. Hibar, N. Jahanshad, Y. Wang, J. Ye, P. Thompson and J. Wang, "Large-scale collaborative imaging genetics studies of risk genetic factors for ad across multiple institutions", in "Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016", (Springer, 2016a), accepted.

Li, Y., J. Wang, T. Yang, J. Chen, L. Liu, P. Thompson and J. Ye, "Detection of Alzheimer's Disease risk factors by tree-structured group Lasso screening", in "Biomedical Imaging (ISBI), 2016 IEEE 13th International Symposium on", (IEEE, 2016b), accepted.

Li, Y., C. J. Willer, J. Ding, P. Scheet and G. R. Abecasis, "Mach: using sequence and genotype data to estimate haplotypes and unobserved genotypes", Genetic epidemiology **34**, 8, 816–834 (2010).

Lin, B., Q. Li, Q. Sun, M.-J. Lai, I. Davidson, W. Fan and J. Ye, "Stochastic coordinate coding and its application for drosophila gene expression pattern annotation", CoRR **abs/1407.8147** (2014).

Liu, H., M. Palatucci and J. Zhang, "Blockwise coordinate descent procedures for the multi-task lasso, with applications to neural semantic basis discovery", in "Proceedings of the 26th Annual International Conference on Machine Learning", ICML '09, pp. 649–656 (ACM, New York, NY, USA, 2009a), URL `http://doi.acm.org/10.1145/1553374.1553458`.

Liu, J., *Penalized methods in genome-wide association studies*, Ph.D. thesis (2011).

Liu, J., J. Huang, S. Ma and K. Wang, "Incorporating group correlations in genome-wide association studies using smoothed group lasso", Biostatistics **14**, 2, 205–219 (2013).

Liu, J., S. Ji and J. Ye, *SLEP: Sparse Learning with Efficient Projections*, Arizona State University, URL `http://www.public.asu.edu/~jye02/Software/SLEP` (2009b).

Liu, J., K. Wang, S. Ma and J. Huang, "Regularized regression method for genome-wide association studies", BMC Proceedings **5**, 9, 1–5, URL `http://dx.doi.org/10.1186/1753-6561-5-S9-S67` (2011).

Liu, J. and J. Ye, "Moreau-yosida regularization for grouped tree structure learning", in "Advances in Neural Information Processing Systems 23", edited by J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel and A. Culotta, pp. 1459–1467 (Curran Associates, Inc., 2010), URL `http://papers.nips.cc/paper/3931-moreau-yosida-regularization-for-grouped-tree-structure-learning.pdf`.

Liu, J., L. Yuan and J. Ye, "An efficient algorithm for a class of fused lasso problems", in "Proceedings of the 16th ACM SIGKDD International Conf. on Knowledge Discovery and Data Mining", pp. 323–332 (ACM, 2010), URL `http://doi.acm.org/10.1145/1835804.1835847`.

Logue, M. W. *et al.*, "A comprehensive genetic association study of Alzheimer disease in african americans", Archives of neurology **68**, 12, 1569–1579 (2011).

Lowe, D. G., "Object recognition from local scale-invariant features", in "Computer vision, 1999. The proceedings of the seventh IEEE international conference on", vol. 2, pp. 1150–1157 (Ieee, 1999).

Lyall, D. M. *et al.*, "Alzheimer's disease susceptibility genes APOE and TOMM40, and brain white matter integrity in the lothian birth cohort 1936", Neurobiology of aging **35**, 6, 1513–e25 (2014).

Maruszak, A. *et al.*, "TOMM40 rs10524523 polymorphism's role in late-onset Alzheimer's disease and in longevity", Journal of Alzheimer's Disease **28**, 2, 309–322 (2012).

McCallum, A., "Multi-label text classification with a mixture model trained by em", in "AAAI99 workshop on text learning", pp. 1–7 (1999).

Medland, S. E., N. Jahanshad, B. M. Neale and P. M. Thompson, "Whole-genome analyses of whole-brain data: working within an expanded search space", Nature neuroscience **17**, 6, 791–800 (2014).

Meier, L., S. Geer and P. Bühlmann, "The group lasso for logistic regression", Journal of the Royal Statistical Society: Series B **70**, 53–71 (2008).

Meinshausen, N. and P. Bühlmann, "Stability selection", Journal of the Royal Statistical Society: Series B (Statistical Methodology) **72**, 4, 417–473 (2010).

Meyer-Lindenberg, A., "The future of fmri and genetics research", NeuroImage **62**, 2, 1286–1292 (2012).

Mu, Y. and F. Gage, "Adult hippocampal neurogenesis and its role in Alzheimers disease", Molecular Neurodegeneration **6**, 85 (2011).

Negahban, S. and M. Wainwright, "Joint support recovery under high-dimensional scaling: Benefits and perils of $\ell_{1,\infty}$-regularization", in "Advances in Neural Information Processing Systems", pp. 1161–1168 (2008).

Nesterov, Y., "A method of solving a convex programming problem with convergence rate o (1/k2)", Soviet Mathematics Doklady **27**, 2, 372–376 (1983).

Nesterov, Y., "Gradient methods for minimizing composite objective function", CORE Discussion Papers 2007076, Universit catholique de Louvain, Center for Operations Research and Econometrics (CORE), URL `http://EconPapers.repec.org/RePEc:cor:louvco:2007076` (2007).

Obozinski, G., B. Taskar and M. I. Jordan, "Joint covariate selection for grouped classification", Tech. rep., Statistics Department, UC Berkeley (2007).

Olshausen, B. A. *et al.*, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images", Nature **381**, 6583, 607–609 (1996).

Peng, J., J. Zhu, A. Bergamaschi, W. Han, D. Noh, J. Pollack and P. Wang, "Regularized multivariate regression for indentifying master predictors with application to integrative genomics study of breast cancer", The Annals of Appliced Statistics **4**, 53–77 (2010).

Potkin, S. G., G. Guffanti, A. Lakatos, J. A. Turner, F. Kruggel, J. H. Fallon, A. J. Saykin, A. Orro, S. Lupoli, E. Salvi *et al.*, "Hippocampal atrophy as a quantitative trait in a genome-wide association study identifying novel susceptibility genes for Alzheimers disease", PloS one **4**, 8, e6501 (2009).

Pritchard, J. K. and M. Przeworski, "Linkage disequilibrium in humans: models and data", The American Journal of Human Genetics **69**, 1, 1–14 (2001).

Psaty, B. M., C. J. O'Donnell, V. Gudnason, K. L. Lunetta, A. R. Folsom, J. I. Rotter, A. G. Uitterlinden, T. B. Harris, J. C. Witteman, E. Boerwinkle *et al.*, "Cohorts for heart and aging research in genomic epidemiology (charge) consortium design of prospective meta-analyses of genome-wide association studies from 5 cohorts", Circulation: Cardiovascular Genetics **2**, 1, 73–80 (2009).

Purcell, S., B. Neale, K. Todd-Brown, L. Thomas, M. A. Ferreira, D. Bender, J. Maller, P. Sklar, P. I. De Bakker, M. J. Daly *et al.*, "Plink: a tool set for whole-genome association and population-based linkage analyses", The American Journal of Human Genetics **81**, 3, 559–575 (2007).

Quattoni, A., X. Carreras, M. Collins and T. Darrell, "An efficient projection for $\ell_{1,\infty}$,infinity regularization", in "Proceedings of the 26th Annual International Conference on Machine Learning", ICML '09, pp. 857–864 (ACM, New York, NY, USA, 2009), URL `http://doi.acm.org/10.1145/1553374.1553484`.

Ranzato, M., "Tutorial on deep learning for vision", A tutorial in conjunction with the Intl. Conference in Computer Vision (CVPR) 2014., URL `https://sites.google.com/site/deeplearningcvpr2014/` (2014).

Read, J., "A pruned problem transformation method for multi-label classification", in "Proc. 2008 New Zealand Computer Science Research Student Conference (NZCSRS 2008)", vol. 143150 (2008).

Reilly, K. M., "Brain tumor susceptibility: the role of genetic factors and uses of mouse models to unravel risk", Brain pathology **19**, 1, 121–131 (2009).

Reuter, M., N. J. Schmansky, H. D. Rosas and B. Fischl, "Within-subject template estimation for unbiased longitudinal image analysis", NeuroImage **61**, 4, 1402–1418 (2012).

Rockafellar, R. T., *Convex Analysis*, chap. Theorem 25.1, p. 242 (Princeton University Press, 1970).

Schapire, R. E. and Y. Singer, "Boostexter: A boosting-based system for text categorization", Machine learning **39**, 2, 135–168 (2000).

Schuff, N., N. Woerner, L. Boreta, T. Kornfield, L. Shaw, J. Trojanowski, P. Thompson, C. Jack, M. Weiner, D. N. Initiative *et al.*, "Mri of hippocampal volume loss in early Alzheimer's disease in relation to ApoE genotype and biomarkers", Brain **132**, 4, 1067–1077 (2009).

Shaw, P., J. P. Lerch, J. C. Pruessner, K. N. Taylor, A. B. Rose, D. Greenstein, L. Clasen, A. Evans, J. L. Rapoport and J. N. Giedd, "Cortical morphology in children and adolescents with different apolipoprotein e gene polymorphisms: an observational study", The Lancet Neurology **6**, 6, 494–500 (2007).

Sherry, S. T., M.-H. Ward, M. Kholodov, J. Baker, L. Phan, E. M. Smigielski and K. Sirotkin, "dbsnp: the ncbi database of genetic variation", Nucleic acids research **29**, 1, 308–311 (2001).

Silla Jr, C. N. and A. A. Freitas, "A survey of hierarchical classification across different application domains", Data Mining and Knowledge Discovery **22**, 1-2, 31–72 (2011).

Simon, N., J. Friedman., T. Hastie. and R. Tibshirani, "A Sparse-Group Lasso", Journal of Computational and Graphical Statistics **22**, 231–245 (2013).

Singh, M., P. Singh, P. Juneja, S. Singh and T. Kaur, "Snpsnp interactions within APOE gene influence plasma lipids in postmenopausal osteoporosis", Rheumatology International **31**, 421–423 (2011).

Sorower, M. S., "A literature survey on algorithms for multi-label learning", Oregon State University, Corvallis (2010).

Sprechmann, P., I. Ramírez, G. Sapiro. and Y. Eldar, "C-HiLasso: a collaborative hierarchical sparse modeling framework", IEEE Transactions on Signal Processing **59**, 4183–4198 (2011).

Spyromitros, E., G. Tsoumakas and I. Vlahavas, "An empirical study of lazy multilabel classification algorithms", in "Artificial Intelligence: Theories, Models and Applications", pp. 401–406 (Springer, 2008).

Srivastava, N., G. E. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting.", Journal of Machine Learning Research **15**, 1, 1929–1958 (2014).

Stein, J. L., S. E. Medland, A. A. Vasquez, D. P. Hibar, R. E. Senstad, A. M. Winkler, R. Toro, K. Appel, R. Bartecek, Ø. Bergmann *et al.*, "Identification of common variants associated with human hippocampal and intracranial volumes", Nature genetics **44**, 5, 552–561 (2012).

Subramanian, A., P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander and J. P. Mesirov, "Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles", Proceedings of the National Academy of Sciences **102**, 43, 15545–15550, URL `http://www.pnas.org/content/102/43/15545.abstract` (2005).

Sun, D., T. Erp, P. Thompson, C. Bearden, M. Daley, L. Kushan, M. Hardt, K. Nuechterlein, A. Toga and T. Cannon, "Elucidating a magnetic resonance imaging-based neuroanatomic biomarker for psychosis: classification analysis using

probabilistic brain atlas and machine learning algorithms", Biological Psychiatry **66**, 1055–1–60 (2009).

Szlam, A., K. Gregor and Y. LeCun, "Fast approximations to structured sparse coding and applications to object classification", in "Computer Vision–ECCV 2012", pp. 200–213 (Springer, 2012).

Tao, P. D. and L. T. H. An, "Convex analysis approach to dc programming: Theory, algorithms and applications", Acta Mathematica Vietnamica **22**, 1, 289–355 (1997).

Tao, P. D. *et al.*, "Duality in dc (difference of convex functions) optimization. subgradient methods", in "Trends in Mathematical Optimization", pp. 277–293 (Springer, 1988).

Tao, P. D. *et al.*, "The dc (difference of convex functions) programming and dca revisited with dc models of real world nonconvex optimization problems", Annals of Operations Research **133**, 1-4, 23–46 (2005).

The International HapMap Consortium, "The international hapmap project", Nature **426**, 6968, 789–796 (2003).

Thompson, C. L., L. Ng, V. Menon, S. Martinez *et al.*, "A high-resolution spatiotemporal atlas of gene expression of the developing mouse brain", Neuron , 0, –, URL `http://www.sciencedirect.com/science/article/pii/S0896627314004486` (2014).

Thompson, P. M., T. Ge, D. C. Glahn, N. Jahanshad and T. E. Nichols, "Genetics of the connectome", Neuroimage **80**, 475–488 (2013).

Tibshirani, R., "Regression shringkage and selection via the lasso", Journal of the Royal Statistical Society Series B **58**, 267–288 (1996).

Tibshirani, R., J. Bien, J. Friedman, T. Hastie, N. Simon, J. Taylor and R. Tibshirani, "Strong rules for discarding predictors in lasso-type problems", Journal of the Royal Statistical Society Series B **74**, 245–266 (2012).

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

Tibshirani, R. and P. Wang, "Spatial smoothing and hot spot detection for cgh data using the fused lasso", Biostatistics **9**, 1, 18–29 (2008).

Tsoumakas, G. and I. Katakis, "Multi-label classification: An overview", Dept. of Informatics, Aristotle University of Thessaloniki, Greece (2006).

Tsoumakas, G., I. Katakis and I. Vlahavas, "Mining multi-label data", in "Data mining and knowledge discovery handbook", pp. 667–685 (Springer, 2010).

Tsoumakas, G. and I. Vlahavas, "Random k-labelsets: An ensemble method for multilabel classification", in "Machine learning: ECML 2007", pp. 406–417 (Springer, 2007).

Tycko, B. *et al.*, "APOE and APOC1 promoter polymorphisms and the risk of Alzheimer disease in african american and caribbean hispanic individuals", Archives of neurology **61**, 9, 1434–1439 (2004).

Vedaldi, A. and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms", `http://www.vlfeat.org/` (2008).

Vidyasagar, M., "Machine learning methods in the computational biology of cancer", Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences **471**, 2173, URL `http://rspa.royalsocietypublishing.org/content/471/2173/20140805` (2014).

Vinje, W. and J. Gallant, "Sparse coding and decorrelation in primary visual cortex during natural vision", Science **287**, 1273–1276 (2000).

Waegeman, W., E. Hllermeier and K. Dembczyski, "Icml tutorial on multi-target prediction", URL `http://www.ngdata.com/icml-2013-tutorial-multi-target-prediction/` (2013).

Wang, H., C. Ding and H. Huang, "Multi-label linear discriminant analysis", in "Computer Vision–ECCV 2010", pp. 126–139 (Springer, 2010).

Wang, H., M. Huang and X. Zhu, "A generative probabilistic model for multi-label classification", in "Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on", pp. 628–637 (IEEE, 2008).

Wang, H., F. Nie, H. Huang, S. L. Risacher, A. J. Saykin, L. Shen *et al.*, "Identifying disease sensitive and quantitative trait-relevant biomarkers from multidimensional heterogeneous imaging genetics data via sparse multimodal multitask learning", Bioinformatics **28**, 12, i127–i136 (2012).

Wang, J., W. Fan and J. Ye, "Fused lasso screening rules via the monotonicity of subdifferentials", IEEE Transactions on Pattern Analysis and Machine Intelligence **PP**, 99, 1–1 (2015a).

Wang, J., J. Liu and J. Ye, "Efficient mixed-norm regularization: Algorithms and safe screening methods", CoRR **abs/1307.4156**, URL `http://arxiv.org/abs/1307.4156` (2013).

Wang, J., P. Wonka and J. Ye, "Lasso screening rules via dual polytope projection", Journal of Machine Learning Research **16**, 1063–1101 (2015b).

Wang, J., T. Yang, P. Thompson and J. Ye, "Sparse models for imaging genetics", in "Machine Learning and Medical Imaging", edited by G. Wu, D. Shen and M. R. Sabuncu, pp. 129 – 151 (Academic Press, 2016a), URL `http://www.sciencedirect.com/science/article/pii/B9780128040768000050`.

Wang, J. and J. Ye, "Two-layer feature reduction for sparse-group lasso via decomposition of convex sets", in "Advances in Neural Information Processing Systems 27", edited by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence

and K. Q. Weinberger, pp. 2132–2140 (Curran Associates, Inc., 2014), URL http://papers.nips.cc/paper/5327-two-layer-feature-reduction-for-sparse-group-lasso-via-decomposition-of-convex-sets.pdf.

Wang, J. and J. Ye, "Multi-layer feature reduction for tree structured group lasso via hierarchical projection", in "Advances in Neural Information Processing Systems 28", edited by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama and R. Garnett, pp. 1279–1287 (Curran Associates, Inc., 2015), URL http://papers.nips.cc/paper/5838-multi-layer-feature-reduction-for-tree-structured-group-lasso-via-hierarchical-projection.pdf.

Wang, J., J. Zhou, J. Liu, P. Wonka and J. Ye, "A safe screening rule for sparse logistic regression", in "Advances in Neural Information Processing Systems 27", edited by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence and K. Q. Weinberger, pp. 1053–1061 (Curran Associates, Inc., 2014), URL http://papers.nips.cc/paper/5294-a-safe-screening-rule-for-sparse-logistic-regression.pdf.

Wang, Y., S. Wang, J. Tang, H. Liu and B. Li, "PPP: Joint pointwise and pairwise image label prediction", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", (2016b).

Warde-Farley, D., S. L. Donaldson, O. Comes, K. Zuberi, R. Badrawi, P. Chao, M. Franz, C. Grouios, F. Kazi, C. T. Lopes *et al.*, "The genemania prediction server: biological network integration for gene prioritization and predicting gene function", Nucleic acids research **38**, suppl 2, W214–W220 (2010).

Weiner, M., M. Albert, C. DeCarli, S. de Kosky, M. de Leon, N. Foster, N. Fox, R. Frank, R. Frackowiak, C. Jack *et al.*, "The use of mri and pet for clinical diagnosis of dementia and investigation of cognitive impairment: A consensus report", Alzheimers Association, Chicago, IL (2005).

World Health Organization, "World cancer report 2014.", (2014).

Wright, J., Y. Ma, J. Mairal, G. Sapiro, T. Huang and S. Yan, "Sparse representation for computer vision and pattern recognition", in "Proceedings of IEEE", (2010).

Wright, S., R. Nowak and M. Figueiredo, "Sparse reconstruction by separable approximation", IEEE Transactions on Signal Processing **57**, 7, 2479–2493 (2009).

Wu, T. T., Y. F. Chen, T. Hastie, E. Sobel and K. Lange, "Genomewide association analysis by lasso penalized logistic regression", Bioinformatics **25**, 714–721 (2009).

Yang, J., T. Ferreira, A. Morris, S. Medland, P. Madden, A. Heath, N. Martin, G. Montgomery, M. Weedon, R. Loos, T. Frayling, M. McCarthy, J. Hirschhorn, M. Goddard and P. Visscher, "Conditional and joint multiple-SNP analysis of GWAS summary statistics identifies additional variants influencing complex traits", Nature genetics **44**, 369–375 (2012a).

Yang, S., Z. Lu, X. Shen, P. Wonka and J. Ye, "Fused multiple graphical lasso", SIAM Journal on Optimization **25**, 2, 916–943 (2015a).

Yang, S., L. Yuan, Y.-C. Lai, X. Shen, P. Wonka and J. Ye, "Feature grouping and selection over an undirected graph", in "Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 922–930 (ACM, 2012b).

Yang, T., J. Liu, P. Gong, R. Zhang, X. Shen and J. Ye, "Absolute fused lasso & its application to genome-wide association studies", in "Proceedings of the 22th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", (ACM, 2016), accepted.

Yang, T., J. Wang, Q. Sun, D. P. Hibar, N. Jahanshad, L. Liu, Y. Wang, L. Zhan, P. M. Thompson and J. Ye, "Detecting genetic risk factors for Alzheimer's disease in whole genome sequence data via lasso screening", in "Biomedical Imaging (ISBI), 2015 IEEE 12th International Symposium on", pp. 985–989 (IEEE, 2015b).

Yang, T., X. Zhao, B. Lin, T. Zeng, S. Ji and J. Ye, "Automated gene expression pattern annotation in the mouse brain", Pacific Symposium on Biocomputing p. 144 155 (2015c).

Ye, J. and J. Liu, "Sparse methods for biomedical data", ACM SIGKDD Explorations Newsletter **14**, 1, 4–15 (2012).

Yogatama, D. and N. A. Smith, "Linguistic structured sparsity in text categorization", in "Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)", pp. 786–796 (Association for Computational Linguistics, Baltimore, Maryland, 2014), URL `http://www.aclweb.org/anthology/P14-1074`.

Yu, K., S. Yu and V. Tresp, "Multi-label informed latent semantic indexing", in "Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval", pp. 258–265 (ACM, 2005).

Yuan, M. and Y. Lin, "Model selection and estimation in regression with grouped variables", Journal of the Royal Statistical Society Series B **68**, 49–67 (2006).

Zeng, T. and S. Ji, "Automated annotation of gene expression patterns in the developing mouse brain atlas", private communication (2014).

Zhang, M.-L. and Z.-H. Zhou, "A review on multi-label learning algorithms", Knowledge and Data Engineering, IEEE Transactions on **26**, 8, 1819–1837 (2014).

Zhang, X., Q. Yuan, S. Zhao, W. Fan, W. Zheng and Z. Wang, "Multi-label classification without the multi-label cost.", in "SDM", vol. 10, pp. 778–789 (SIAM, 2010).

Zhao, P., G. Rocha and B. Yu, "The composite absolute penalties family for grouped and hierarchical variable selection", The Annals of Statistics pp. 3468–3497 (2009).

Zhao, P. and B. Yu, "On model selection consistency of lasso", Journal of Machine Learning Research **7**, 2541–2563 (2006).

Zhou, J., J. Chen and J. Ye, *MALSAR: Multi-tAsk Learning via StructurAl Regularization*, Arizona State University, URL `http://www.public.asu.edu/~jye02/Software/MALSAR` (2011).

Zhou, Q., F. Zhao, Z.-p. Lv, C.-g. Zheng, W.-d. Zheng, L. Sun, N.-n. Wang, S. Pang, F. M. de Andrade, M. Fu *et al.*, "Association between APOC1 polymorphism and Alzheimers disease: A case-control study and meta-analysis", PloS one **9**, 1, e87017 (2014).

Zhu, J. and T. Hastie, "Classification of gene microarrays by penalized logistic regression", Biostatistics **5**, 427–443 (2004).

Zhu, S., X. Ji, W. Xu and Y. Gong, "Multi-labelled classification using maximum entropy method", in "Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval", pp. 274–281 (ACM, 2005).

Zhu, Y., X. Shen and W. Pan, "Simultaneous grouping pursuit and feature selection over an undirected graph", Journal of the American Statistical Association **108**, 502, 713–725 (2013).

Zou, H. and T. Hastie, "Regularization and variable selection via the elastic net", Journal of the Royal Statistical Society: Series B (Statistical Methodology) **67**, 2, 301–320, URL `http://dx.doi.org/10.1111/j.1467-9868.2005.00503.x` (2005).

# APPENDIX A

# DC PROGRAMMING FOR SOLVING AFL

The AFL formulation in Eq. (4.11) is a non-convex optimization problem. We propose to use the DC programming to solve it. By noting that

$$||x_i| - |x_{i+1}|| = |x_i + x_{i+1}| + |x_i - x_{i+1}| - (|x_i| + |x_{i+1}|),$$

we decompose the objective function in Eq. (4.11) into the difference of the following two functions:

$$f_1(\mathbf{x}) = \text{loss}(\mathbf{x}) + \lambda_1\|\mathbf{x}\|_1 + \lambda_2 \sum_{i=1}^{p-1} (|x_i + x_{i+1}| + |x_i - x_{i+1}|),$$

$$f_2(\mathbf{x}) = \lambda_2 \sum_{i=1}^{p-1} (|x_i| + |x_{i+1}|).$$

Denote the affine minorization of $f_2(\mathbf{x})$ as $f_2^k(\mathbf{x}) = f_2(\mathbf{x}^k) + \langle \mathbf{x} - \mathbf{x}^k, \partial f_2(\mathbf{x}^k) \rangle$, where $\langle \cdot, \cdot \rangle$ refers to the inner product. Then the DC programming solves problem (4.11) by iteratively solving:

$$\min_{\mathbf{x} \in \mathbb{R}^p} f_1(\mathbf{x}) - f_2^k(\mathbf{x}). \tag{A.1}$$

Since $\langle \mathbf{x}^k, \partial f_2(\mathbf{x}^k) \rangle$ is a constant, problem (A.1) is equivalent to:

$$\min_{\mathbf{x} \in \mathbb{R}^p} f_1(\mathbf{x}) - \langle \mathbf{x}, \partial f_2(\mathbf{x}^k) \rangle. \tag{A.2}$$

and let $\mathbf{c}^k = \partial f_2(\mathbf{x}^k)$, problem (A.1) can be rewritten as:

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^p} & \text{loss}(\mathbf{x}) - (\mathbf{c}^k)^T \mathbf{x} \\ & + \lambda_1\|\mathbf{x}\|_1 + \lambda_2 \sum_{i=1}^{p-1} (|x_i + x_{i+1}| + |x_i - x_{i+1}|). \end{aligned} \tag{A.3}$$

Note that

$$c_i^k = \lambda_2 d_i \,\text{sgn}\,(x_i^k), \tag{A.4}$$

where $d_1 = d_p = 1, d_i = 2, 2 \le i \le p-1$; $\text{sgn}(\cdot)$ is the signum function. In addition, since $\max(|x_i|, |x_{i+1}|) = \frac{1}{2}(|x_i + x_{i+1}| + |x_i - x_{i+1}|)$, problem (A.3) is equivalent to

$$\min_{\mathbf{x} \in \mathbb{R}^p} \text{loss}(\mathbf{x}) - (\mathbf{c}^k)^T \mathbf{x} + \lambda_1\|\mathbf{x}\|_1 + 2\lambda_2 \sum_{i=1}^{p-1} \max(|x_i|, |x_{i+1}|).$$

# APPENDIX B

# PROOF FOR LEMMA 1 – PROPERTIES OF THE PROXIMAL OPERATOR

**Proof:** We prove these properties of the proximal operator problem (4.20) as follows.

i) If $u_i \geq 0$ and $x_i^* < 0$, we can construct $\tilde{\mathbf{x}}^*$ as follows:

$$\tilde{x}_i^* = 0, \tilde{x}_j^* = x_j^*, \forall j \neq i.$$

It can easily be shown that $\phi(\tilde{\mathbf{x}}^*) < \phi(\mathbf{x}^*)$. This contradicts with the fact that $\mathbf{x}^*$ is the minimizer to (4.20). If $u_i \geq 0$ and $x_i^* > u_i$, we can construct $\tilde{\mathbf{x}}^*$ as follows:

$$\tilde{x}_i^* = u_i, \tilde{x}_j^* = x_j^*, \forall j \neq i.$$

It can easily be shown that $\phi(\tilde{\mathbf{x}}^*) < \phi(\mathbf{x}^*)$. This contradicts with the fact that $\mathbf{x}^*$ is the minimizer to (4.20).

ii) This property can be proved in a similar way as i).

iii) Let $\tilde{\mathbf{x}}^* = \pi_\lambda(|\mathbf{u}|)$. We have

$$\begin{aligned}
\phi(\operatorname{sgn}(\mathbf{u}) \odot \tilde{\mathbf{x}}^*) &= \frac{1}{2}\| \operatorname{sgn}(\mathbf{u}) \odot \tilde{\mathbf{x}}^* - \mathbf{u}\|^2 \\
&\quad + \lambda \sum_{i=1}^{p-1} \max(|\operatorname{sgn}(u_i)\tilde{x}_i^*|, |\operatorname{sgn}(u_{i+1})\tilde{x}_{i+1}^*|) \\
&= \frac{1}{2}\| \operatorname{sgn}(\mathbf{u}) \odot (\tilde{\mathbf{x}}^* - |\mathbf{u}|)\|^2 \\
&\quad + \lambda \sum_{i=1}^{p-1} \max(|\tilde{x}_i^*|, |\tilde{x}_{i+1}^*|) \\
&= \frac{1}{2}\|\tilde{\mathbf{x}}^* - |\mathbf{u}|\|^2 + \lambda \sum_{i=1}^{p-1} \max(|\tilde{x}_i^*|, |\tilde{x}_{i+1}^*|).
\end{aligned}$$

Since $\tilde{\mathbf{x}}^* = \pi_\lambda(|\mathbf{u}|)$ and the minimizer is unique, it follows that $\operatorname{sgn}(\mathbf{u}) \odot \tilde{\mathbf{x}}^*$ needs to minimize $\phi(\mathbf{x})$.

iv) We only focus on the case $u_i \geq u_{i+1} \geq 0$ in the proof and the results can be generated to the rest the cases using property iii). With properties i) and ii), we have $u_i \geq x_i^* \geq 0$ and $u_{i+1} \geq x_{i+1}^* \geq 0$. If this property does not hold, we have:

$$u_i \geq u_{i+1} \geq x_{i+1}^* > x_i^* \geq 0. \tag{B.1}$$

Next, we show that $x_{i+1}^* > x_i^*$ leads to a contradiction. With a non-negative $\epsilon$ and assuming $x_{i+1}^* - \epsilon > x_i^*$, we construct $\bar{\mathbf{x}}^*$ and $\tilde{\mathbf{x}}^*$ as follows:

$$\bar{x}_{i+1}^* = x_{i+1}^* - \epsilon, \bar{x}_j^* = x_j^*, \forall j \neq i+1, \tag{B.2}$$

$$\tilde{x}_i^* = x_{i+1}^* + \epsilon, \tilde{x}_j^* = x_j^*, \forall j \neq i. \tag{B.3}$$

where the $i+1$ entry of $\mathbf{x}^*$ is decreased by $\epsilon$ in constructing $\bar{\mathbf{x}}^*$ and the $i$ entry of $\mathbf{x}^*$ is increased by $\epsilon$ in constructing $\tilde{\mathbf{x}}^*$. Denote

$$d = -\frac{1}{2}(x_{i+1}^* - u_{i+1})^2 + \frac{1}{2}(x_{i+1}^* - \epsilon - u_{i+1})^2.$$

If $x^*_{i+1} < x^*_{i+2}$, we have

$$\phi(\bar{\mathbf{x}}) - \phi(\mathbf{x}^*) = d - \lambda\epsilon. \tag{B.4}$$

If $x^*_{i+1} - \epsilon > x^*_{i+2}$, we have

$$\phi(\bar{\mathbf{x}}) - \phi(\mathbf{x}^*) = d - 2\lambda\epsilon. \tag{B.5}$$

If $x^*_{i+1} \geq x^*_{i+2} \geq x^*_{i+1} - \epsilon$, we have

$$\phi(\bar{\mathbf{x}}) - \phi(\mathbf{x}^*) \leq d - \lambda\epsilon. \tag{B.6}$$

$$\phi(\bar{\mathbf{x}}) - \phi(\mathbf{x}^*) \geq d - 2\lambda\epsilon. \tag{B.7}$$

In summary, we have

$$\begin{aligned}\phi(\bar{\mathbf{x}}) - \phi(\mathbf{x}^*) \leq & g_1(\epsilon) = -\frac{1}{2}(x^*_{i+1} - u_{i+1})^2 \\ & + \frac{1}{2}(x^*_{i+1} - \epsilon - u_{i+1})^2 - \lambda\epsilon.\end{aligned} \tag{B.8}$$

Similarly, we have

$$\begin{aligned}\phi(\tilde{\mathbf{x}}) - \phi(\mathbf{x}^*) \leq g_2(\epsilon) = & -\frac{1}{2}(x^*_i - u_i)^2 \\ & + \frac{1}{2}(x^*_i + \epsilon - u_i)^2 + \lambda\epsilon.\end{aligned} \tag{B.9}$$

It is hard to directly prove either $g_1(\epsilon)$ or $g_2(\epsilon)$ is negative in the case of (B.1). To arrive at the contradiction, we let

$$\begin{aligned}\mathcal{G}(\epsilon) &= g_1(\epsilon) + g_2(\epsilon) \\ &= -\frac{1}{2}(x^*_{i+1} - u_{i+1})^2 + \frac{1}{2}(x^*_{i+1} - \epsilon - u_{i+1})^2 \\ &\quad - \frac{1}{2}(x^*_i - u_i)^2 + \frac{1}{2}(x^*_i + \epsilon - u_i)^2\end{aligned} \tag{B.10}$$

The derivative of $\mathcal{G}(\epsilon)$ is

$$\mathcal{G}'(\epsilon) = 2\epsilon + (u_{i+1} - u_i) + (x^*_i - x^*_{i+1}). \tag{B.11}$$

Making use of (B.1), we can arrive at $\mathcal{G}'(\epsilon) < 0$ when

$$\epsilon \in (0, \frac{x^*_{i+1} - x^*_i}{2}). \tag{B.12}$$

For any $\epsilon$ satisfying (B.12), we have $\mathcal{G}(\epsilon) < 0$, since $\mathcal{G}(0) = 0$ and $\mathcal{G}'(\epsilon) < 0$. Therefore, there exists $\epsilon$ that satisfies (B.12). Hence

$$(\phi(\bar{\mathbf{x}}^*) - \phi(\mathbf{x}^*)) + (\phi(\tilde{\mathbf{x}}^*) - \phi(\mathbf{x}^*)) < 0. \tag{B.13}$$

This leads to the fact that at least one of the following two inequalities holds:

$$(\phi(\bar{\mathbf{x}}^*) - \phi(\mathbf{x}^*)) < 0,$$

$$(\phi(\tilde{\mathbf{x}}^*) - \phi(\mathbf{x}^*)) < 0.$$

This contradicts with the fact that $\mathbf{x}^*$ is the minimizer to (4.20). Therefore, we cannot have $x^*_{i+1} > x^*_i$ in the case $u_i \geq u_{i+1} \geq 0$.

v) This property can be proved in a similar way as iv).

This ends the proof to Lemma 1. □

Based on Lemma 1, we also have the following remark that summarizes the properties of $w$:

**Remark.** *$w_i = 2$ indicates $1 < i < p, u_{i-1} < u_i \leq u_{i+1}$.*
*$w_i = 1$ holds in one of the following four cases:*
*1) $i = 1, u_1 \geq u_2$;*
*2) $i = p, u_{p-1} < u_p$;*
*3) $1 < i < p, u_i \geq u_{i+1}, u_i \leq u_{i-1}$;*
*4) $1 < i < p, u_i < u_{i+1}, u_i > u_{i-1}$.*
*$w_i = 0$ holds in one of the following three cases:*
*1) $i = 1, u_1 < u_2$;*
*2) $i = p, u_{p-1} \geq u_p$;*
*3) $1 < i < p, u_i < u_{i+1}, u_i \geq u_{i-1}$.*
*In addition, it is easy to get that $\sum_{i=1}^{p} w_i = p - 1$.*

# APPENDIX C

# PROOF FOR THEOREM 2 IN SOLVING AFL PROBLEM VIA EUCLIDEAN PROJECTION

**Proof:** According to Lemma 1 i) and ii), we have that the optimal solution to (4.20) is non-negative, i.e., $\mathbf{x}^* \geq 0$. Incorporating the definition of $R$ in (4.21) and Lemma 1 iv) and v), we have $R\mathbf{x}^* \leq 0$. Therefore, we have $\mathbf{x}^* \in P$, where $P$ is defined in (4.24). It is easy to verify that $P$ is a closed convex and nonempty polyhedron. Thus $\mathbf{x}^* = \pi_\lambda(\mathbf{u})$ is the optimal solution to

$$\min_{\mathbf{x}\in P} \left\{ \frac{1}{2}\|\mathbf{x}-\mathbf{u}\|^2 + \lambda \sum_{i=1}^{p-1} \max(|x_i|, |x_{i+1}|) \right\}.$$

Making use of the definitions of $R$ and $\mathbf{w}$ in (4.21) and (4.22), $\forall \mathbf{x} \in P$, we have

$$\sum_{i=1}^{p-1} \max(|x_i|, |x_{i+1}|) = \sum_{i=1}^{p} w_i x_i.$$

Therefore, $\mathbf{x}^* = \pi_\lambda(\mathbf{u})$ is the optimal solution to the problem:

$$\min_{\mathbf{x}\in P} \left\{ \frac{1}{2}\|\mathbf{x}-\mathbf{u}\|^2 + \lambda \sum_{i=1}^{p} w_i x_i \right\}. \tag{C.1}$$

Incorporating (4.23), we can easily verify that, $\pi_\lambda^P(\mathbf{v})$, the optimal solution to(4.25) is also the optimal solution to (C.1). Thus, (4.26) holds. □

# APPENDIX D

# PROOF FOR THEOREM 3 IN SOLVING AFL PROBLEM VIA EUCLIDEAN PROJECTION

**Proof:** We prove (4.29) by the technique of KKT optimality conditions.

By introducing the dual variables $\mathbf{w} \in \mathbb{R}^p$ for the inequality $\mathbf{x} \geq 0$, and $\mathbf{z} \in \mathbb{R}^{p-1}$ for the inequality $R\mathbf{x} \leq 0$, we can write the Lagrangian of (4.25) as:

$$\mathcal{L}(\mathbf{x}, \mathbf{w}, \mathbf{z}) = \frac{1}{2}\|\mathbf{x} - \mathbf{v}\|^2 - \mathbf{w}^T\mathbf{x} + \mathbf{z}^T R\mathbf{x}. \tag{D.1}$$

The inequality constraint functions in (4.25) are affine, and thus Slater's condition holds, which indicates strong duality. Let $\bar{\mathbf{x}}$ and $(\bar{\mathbf{w}}, \bar{\mathbf{z}})$ be any primal and dual optimal points with zero gap for (4.25). The KKT optimality conditions require the following necessary and sufficient conditions:

$$\begin{aligned}
\bar{\mathbf{x}} &\geq 0, && \text{(D.2)}\\
R\bar{\mathbf{x}} &\leq 0, && \text{(D.3)}\\
\bar{\mathbf{w}} &\geq 0, && \text{(D.4)}\\
\bar{\mathbf{z}} &\geq 0, && \text{(D.5)}\\
\bar{\mathbf{x}} &= \mathbf{v} + \bar{\mathbf{w}} - R\bar{\mathbf{z}}. && \text{(D.6)}
\end{aligned}$$

Following a similar analysis, we introduce the dual variable $\mathbf{z} \in \mathbb{R}^{p-1}$ for the inequality $R\mathbf{x} \leq 0$, and write the Lagrangian of (4.28) as:

$$\mathcal{L}(\mathbf{x}, \mathbf{z}) = \frac{1}{2}\|\mathbf{x} - \mathbf{v}\|^2 + \mathbf{z}^T R\mathbf{x}. \tag{D.7}$$

Let $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{z}}$ be any primal and dual optimal points with zero gap for (4.28). The KKT optimality conditions requires the following necessary and sufficient conditions:

$$\begin{aligned}
R\tilde{\mathbf{x}} &\leq 0, && \text{(D.8)}\\
\tilde{\mathbf{z}} &\geq 0, && \text{(D.9)}\\
\tilde{\mathbf{x}} &= \mathbf{v} - R\tilde{\mathbf{z}}. && \text{(D.10)}
\end{aligned}$$

Let

$$\begin{aligned}
\mathbf{x}^* &= \max(\tilde{\mathbf{x}}, 0), && \text{(D.11)}\\
\mathbf{w}^* &= \max(\tilde{\mathbf{x}}, 0) - \tilde{\mathbf{x}}, && \text{(D.12)}\\
\mathbf{z}^* &= \tilde{\mathbf{z}}. && \text{(D.13)}
\end{aligned}$$

Next, we show that $\mathbf{x}^*$ and $(\mathbf{w}^*, \mathbf{z}^*)$ satisfy the KKT conditions (D.3)-(D.6). It is easy to verify the relationships in formulations (D.3), (D.5)-(D.6). $R\mathbf{x}^* \leq 0$ holds as: 1) $R\tilde{\mathbf{x}} \leq 0$, 2) $\mathbf{x}^* = \max(\tilde{\mathbf{x}}, 0)$, and 3) each row of $R$ only contains two entries 1 and -1. As the objectives of (4.25) and (4.28) are strictly convex, $\bar{\mathbf{x}}$ and $\tilde{\mathbf{x}}$ are both unique. Therefore, it follows from (D.12) that (4.29) holds. □

# APPENDIX E

## BASIC INFORMATION OF SELECTED GENES

**Table E.1:** Basic information of selected genes

| | Symbol | Assembly | Chr | Location | # of loci |
|---|---|---|---|---|---|
| AD Candidate Genes | LDLR | GRCh37.p13 | 19 | 11200037..11244506 | 135 |
| | GAPDHS | GRCh37.p13 | 19 | 36024314..36036221 | 22 |
| | BCAM | GRCh37.p13 | 19 | 45312316..45324678 | 15 |
| | PVRL2 | GRCh37.p13 | 19 | 45349393..45392485 | 164 |
| | TOMM40 | GRCh37.p13 | 19 | 45394477..45406946 | 38 |
| | APOE | GRCh37.p13 | 19 | 45409039..45412650 | 5 |
| | APOC1 | GRCh37.p13 | 19 | 45417577..45422606 | 14 |
| | APOC4 | GRCh37.p13 | 19 | 45445495..45448753 | 7 |
| | EXOC3L2 | GRCh37.p13 | 19 | 45715879..45737469 | 88 |
| | CD33 | GRCh37.p13 | 19 | 51728335..51743274 | 16 |
| Associated Genes | LDLRAP1 | GRCh37.p13 | 1 | 25870071..25895377 | 28 |
| | PVRL3 | GRCh37.p13 | 3 | 110790606..110913017 | 73 |
| | APOA5 | GRCh37.p13 | 11 | 116660086..116663136 | 7 |
| | APOA1 | GRCh37.p13 | 11 | 116706467..116708338 | 5 |
| | CRTAM | GRCh37.p13 | 11 | 122709255..122743347 | 75 |
| | GAPDH | GRCh37.p13 | 12 | 6643585..6647537 | 10 |
| | LIPC | GRCh37.p13 | 15 | 58702953..58861073 | 481 |
| | CD226 | GRCh37.p13 | 18 | 67530192..67624412 | 149 |
| | APOC2 | GRCh37.p13 | 19 | 45449239..45452822 | 17 |
| | SOD1 | GRCh37.p13 | 21 | 33031935..33041244 | 15 |

Note that, the location information of genes (start / end locations on chromosome) is obtained from dbSNP [Sherry *et al.* (2001)].