Simultaneous Variable and Feature Group Selection in Heterogeneous Learning:

Optimization and Applications

by

Shuo Xiang

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

Approved November 2014 by the
Graduate Supervisory Committee:

Jieping Ye, Chair
Hans D. Mittelmann
Hasan Davulcu
Jingrui He

ARIZONA STATE UNIVERSITY

December 2014
ABSTRACT

Advances in data collection technologies have made it cost-effective to obtain heterogeneous data from multiple data sources. Very often, the data are of very high dimension and feature selection is preferred in order to reduce noise, save computational cost and learn interpretable models. Due to the multi-modality nature of heterogeneous data, it is interesting to design efficient machine learning models that are capable of performing variable selection and feature group (data source) selection simultaneously (a.k.a bi-level selection). In this thesis, I carry out research along this direction with a particular focus on designing efficient optimization algorithms.

I start with a unified bi-level learning model that contains several existing feature selection models as special cases. Then the proposed model is further extended to tackle the block-wise missing data, one of the major challenges in the diagnosis of Alzheimer’s Disease (AD). Moreover, I propose a novel interpretable sparse group feature selection model that greatly facilitates the procedure of parameter tuning and model selection. Last but not least, I show that by solving the sparse group hard thresholding problem directly, the sparse group feature selection model can be further improved in terms of both algorithmic complexity and efficiency. Promising results are demonstrated in the extensive evaluation on multiple real-world data sets.
For family and 7761 days’ school life
ACKNOWLEDGEMENT

It goes without saying that this thesis could never be accomplished without the tremendous effort from my advisor Dr. Jieping Ye. It is rare to see the combination of smartness, patience, great vision and productivity, always being open-minded and obliging in one person. It has been my privilege to work closely with him and it is Dr. Ye’s continuous support that gets me through the hard times.

I would like to express my sincere gratitude to my committee members: Dr. Hans D. Mittelmann, Dr. Hasan Davulcu and Dr. Jingrui He, for their guidance, support and encouragement. Dr. Mittelmann’s optimization lectures are the first lessons I take in the US and play a fundamental role in all my thesis work. Dr. Davulcu introduces me to an exciting project that bridges my investigation on optimization with real-world applications. Dr. He’s lectures have always been inspiring and rewarding for my research. I would also like to thank Dr. Yalin Wang for his enlightening suggestions during our collaboration.

My colleagues and friends in Dr. Jieping Ye’s research group are one source of incessant innovation and I benefit from them on a day to day basis. Many thanks to: Jianhui Chen, Liang Sun, Lei Yuan, Rita Chattopadhyay, Jiayu Zhou, Sen Yang, Yashu Liu, Cheng Pan, Qian Sun, Zhi Nie, Qingyang Li, Tao Yang, Jun Liu, Chao Zhang, Zheng Wang, Jie Wang, Binbin Lin, Pinghua Gong and Kefei Liu.

I want to thank all my friends that make the past four years memorable. They are (in alphabetical order) Huiji Gao, Ben He, Fengjie Li, Yuan Li, Gang Liu, Yinchao Liu, Shaohua Qiu, Zhaonan Sun, Jiliang Tang, Ye Xu, Yuliang Yin and Li Zhang.

Last but not least, no word could express my appreciation to my family. Unconditional and endless love from my parents and grandma makes me feel I never walk alone. Thanks Qian for being supportive ever since the day we met.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>LIST OF TABLES</th>
<th>vii</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>ix</td>
</tr>
</tbody>
</table>

## CHAPTER

1. **INTRODUCTION** ... 1

2. **BI-LEVEL LEARNING FOR MULTI-SOURCE COMPLETE DATA** ... 4
   2.1 Introduction ... 4
   2.2 A Unified Feature Learning Model for Multi-source Data ... 4
   2.2.1 Relation to Previous Works ... 6
   2.3 Optimization ... 7
   2.4 Experiments ... 9
   2.5 Summary ... 10

3. **BI-LEVEL MULTI-SOURCE LEARNING WITH HETEROGENEOUS BLOCK-WISE MISSING DATA FOR ALZHEIMER’S DISEASE PREDICTION** ... 11
   3.1 Introduction ... 11
   3.1.1 Related Work ... 12
   3.2 Subjects ... 14
   3.3 Incomplete Source-Feature Selection (iSFS) Model ... 16
   3.3.1 Formulation ... 18
   3.3.2 Optimization ... 19
   3.4 Experiments ... 22
   3.4.1 Capability of Source Selection ... 24
   3.4.2 Benefit of Utilizing Incomplete Data ... 25
   3.4.3 Ensemble Learning Methods ... 26
<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.5.2 Evaluation of Convergence</td>
<td>69</td>
</tr>
<tr>
<td>5.5.3 Simulation Results</td>
<td>70</td>
</tr>
<tr>
<td>5.5.4 Real-world Applications</td>
<td>73</td>
</tr>
<tr>
<td>5.6 Extension to Fused Hard Thresholding Models</td>
<td>74</td>
</tr>
<tr>
<td>5.7 Summary</td>
<td>75</td>
</tr>
<tr>
<td>6 CONCLUSION AND OUTLOOK</td>
<td>81</td>
</tr>
<tr>
<td>6.1 Summary of Contributions</td>
<td>81</td>
</tr>
<tr>
<td>6.2 Future Work</td>
<td>83</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>84</td>
</tr>
<tr>
<td>APPENDIX</td>
<td></td>
</tr>
<tr>
<td>A PROOF OF THEOREM 1</td>
<td>91</td>
</tr>
<tr>
<td>B PROOF OF PROPOSITION 2</td>
<td>94</td>
</tr>
<tr>
<td>C ACCELERATED GRADIENT METHOD</td>
<td>96</td>
</tr>
<tr>
<td>D ALGORITHM FOR SOLVING 4.13</td>
<td>98</td>
</tr>
<tr>
<td>E THE ADMM PROJECTION ALGORITHM</td>
<td>101</td>
</tr>
<tr>
<td>F THE DYKSTRA’S ALGORITHM</td>
<td>104</td>
</tr>
<tr>
<td>G PROOF OF THEOREM 3</td>
<td>106</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Statistics of the ADNI Data Set and the Data Sources Used In Our Evaluations, Where AD, pMCI, sMCI and NC Stand for Alzheimers Disease Patients, Progressive Mild Cognitive Impairment Patients, Stable Mild Cognitive Impairment Patients, and Normal Controls Respectively</td>
<td>16</td>
</tr>
<tr>
<td>3.2 Classification Results of AD Patients Versus Normal Controls with 10% Data for Training. All Results Are Averaged Over 10 Replications</td>
<td>24</td>
</tr>
<tr>
<td>3.3 Classification Results of AD Patients Versus Stable MCI Patients With 10% Data for Training. All Results Are Averaged Over 10 Replications</td>
<td>25</td>
</tr>
<tr>
<td>3.4 Classification Results of Progressive MCI Patients Versus Normal Controls With 10% Data for Training. All Results Are Averaged Over 10 Replications</td>
<td>26</td>
</tr>
<tr>
<td>3.5 Classification Results of AD Patients Versus Normal Controls With 50% Data for Training. All Results Are Averaged Over 10 Replications</td>
<td>27</td>
</tr>
<tr>
<td>3.6 Classification Results of AD Patients Versus Stable MCI Patients With 50% Data for Training. All Results Are Averaged Over 10 Replications</td>
<td>28</td>
</tr>
<tr>
<td>3.7 Classification Results of Progressive MCI Patients Versus Normal Controls With 50% Data for Training. All Results Are Averaged Over 10 Replications</td>
<td>29</td>
</tr>
<tr>
<td>4.1 Running Time (in seconds) of Dykstra’s, ADMM and Our Projection Algorithm. All Three Algorithms Are Averaged Over 100 Replications</td>
<td>49</td>
</tr>
<tr>
<td>4.2 Distance Between the Optimal Solution of Projection Algorithms and That of The CVX. All The Results Are Averaged Over 100 Replications</td>
<td>50</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------------------------------------------------</td>
</tr>
<tr>
<td>4.3</td>
<td>Comparison of Performance on Synthetic Data. All the Results Are</td>
</tr>
<tr>
<td></td>
<td>Averaged for 100 Replications. DC, Which Stands for Difference of</td>
</tr>
<tr>
<td></td>
<td>Convex functions, Denotes Our Proposed Method</td>
</tr>
<tr>
<td>4.4</td>
<td>Comparison of Performance on EEG Data. DC Which Stand for Difference</td>
</tr>
<tr>
<td></td>
<td>of Convex functions, Denotes Our Proposed Method</td>
</tr>
<tr>
<td>5.1</td>
<td>Specific Settings for Each Variant Considered in the Work. The Last</td>
</tr>
<tr>
<td></td>
<td>Two Columns Denote the Lipschitz and Sufficient Decrease Line Search</td>
</tr>
<tr>
<td></td>
<td>Criterion Respectively</td>
</tr>
<tr>
<td>5.2</td>
<td>Experiment Setup for Evaluation of SGHT</td>
</tr>
<tr>
<td>5.3</td>
<td>Comparison of Performance on Synthetic Data. NO, FP and FN Denote</td>
</tr>
<tr>
<td></td>
<td>Number, False Positive Number and False Negative Number Respectively.</td>
</tr>
<tr>
<td></td>
<td>All the Results Are Averaged Over 10 Replications</td>
</tr>
<tr>
<td>5.4</td>
<td>Comparison of Performance on the Boston Housing Data Set. All the</td>
</tr>
<tr>
<td></td>
<td>Results Are Averaged Over 10 Replications</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 An Illustration of an Incomplete Multi-source Data With Three Sources</td>
<td>12</td>
</tr>
<tr>
<td>3.2 Illustration of the Proposed Learning Model</td>
<td>18</td>
</tr>
<tr>
<td>3.3 The Classification Results of iSFS and iMSF on ADNI Data Set With Additional Noisy Data Sources</td>
<td>29</td>
</tr>
<tr>
<td>3.4 ROC Curves Given by iSFS (on Both Complete and Incomplete Data), Lasso and Group Lasso</td>
<td>30</td>
</tr>
<tr>
<td>3.5 ROC Curves of The Ensemble Methods. The Ratio of The Training Set Varies from 25% to 75% and the Performance On Three Tasks: AD vs. NC, AD vs. Stable MCI and Progressive MCI vs. Normal Controls, Are Reported. The Blue Curve Denotes the Majority Voting Approach, and the Linear Regression Ensemble Method is Represented by the Red Curve</td>
<td>30</td>
</tr>
<tr>
<td>3.6 Illustration of the Convergence of Algorithm 1. The X-axis Denotes The Number of Iterations and the Y-axis Denotes the Objective Value of Eq. (3.1)</td>
<td>31</td>
</tr>
<tr>
<td>3.7 Running Time (in seconds) of the Proposed Algorithm With Increasing Number of Samples and Number of Sources on Synthetic Data</td>
<td>31</td>
</tr>
<tr>
<td>4.1 The Average Running Time for Different Algorithms To Achieve the Precision Level Listed in Table 4.2</td>
<td>50</td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>5.1 Illustration of the Order of Computation for Each Element in $T$. While Computing $T(i, j, k)$, We Only Need Values in Those Red Squares, Which Are Located in the Previous Rectangle (in Terms of $i$-Axis) and of Equal or Smaller Coordinates on Axes $j$ and $k$. Therefore the Computation Can be Naturally Carried Out in Three Nested Loops, One for Each Axis Respectively.</td>
<td>64</td>
</tr>
<tr>
<td>5.2 Illustration of the Convergence Behavior of the Proposed Algorithm. The Parameter Estimation Error Decreases Linearly Before Entering into a Region Centered at $x^<em>$ With Radius Proportional to the Prediction Error of $x^</em>$.</td>
<td>67</td>
</tr>
<tr>
<td>5.3 Running Time (in Seconds) of Algorithm 8 in Different Scenarios.</td>
<td>70</td>
</tr>
<tr>
<td>5.4 Convergence Results of Different Variants of the Proposed Discrete Optimization Approach on Synthetic Data, Where ISTA-L and FISTA-C Stand for ISTA with Lipschiz Line Search Criterion and FISTA with Const Step-size Initialization. All The Algorithms Are Evaluated on Four Data Sets, from Top to Bottom, of Which the Size of $A$ Is (100,2000), (100,5000), (1000,20000) and (1000,50000) Respectively. The Number of Selected Group ($s_2$) is Chosen from $0.1</td>
<td>G</td>
</tr>
</tbody>
</table>
5.5 Illustration of the Grouping Effect in the Ground Truth Model $\tilde{x}$. Both cases include redundant groups (Group 7 to Group 20). In addition, the first case contains a bi-level sparsity. The values within each group are identical, as shown in the color map.
Recent advances in data collection technologies have made it possible to collect a large amount of data for many application domains. Very often, these data come from multiple sources. For instance, in the study of Alzheimer’s Disease (AD), different types of measurements such as magnetic resonance imaging (MRI), positron emission tomography (PET), cerebrospinal fluid (CSF), blood test, protein expression data, and genetic data have been collected as they provide complementary information for the diagnosis of AD \cite{Ye2008, Zhang2012}. In bioinformatics, different types of biological data including protein-protein interactions, gene expression and amino sequences have been collected for protein classification \cite{Lanckriet2004}. Extraction of the great wealth of information from such multi-source (a.k.a multi-modality) data has become a crucial step in knowledge discovery. Data mining and machine learning methods have been increasingly used to analyze multi-source data \cite{Troyanskaya2003, Crammer2008, Xu2007}. It is expected that the performance can be significantly improved if information from different sources can be properly integrated and leveraged. Multi-source learning has thus attracted great attentions in various application domains from biomedicalinformatics \cite{Huopaniemi2010, Ye2008} to web mining \cite{Aizawa2005, Xu2007}.

In many applications, the collected data is also of very high dimension, e.g., medical images and gene/protein expression data. However, the high-dimensional data often contains redundant information or even noisy or corrupted entries and thus poses a potential challenge. In order to build a stable and comprehensible learning
model with good generalization capability, feature selection plays a critical role and has been one of the most active research topics in machine learning. Over the past decades, with the development of compressed sensing techniques \cite{Tibshirani1996, Candes2005, Donoho2006}, joint modeling of prediction and feature selection gains its popularity and draws extensive studies \cite{Zou2008, Liu2009b, Bach2011, Zhang2011, Ye2012, Xu2012}. In the meantime, it is also believed that when the data possesses certain grouping structures, selecting feature groups together with individual features can be beneficial \cite{Yuan2006, Wang2007, Breheny2009, Huang2010, Xiang2013a}. In the literature, simultaneous selection of features and feature groups is also referred to as bi-level selection \cite{Huang2010, Xiang2013a} and we will use these two terms interchangeably throughout the thesis.

In addition to the multi-modality and the high dimensionality, the existence of block-wise missing data poses significant difficulties in many applications, especially those from biomedical area. Traditionally, missing data in machine learning tasks are handled by estimating the unknown values based on the observed ones (a.k.a. imputation). However this approach neglects the block-wise missing pattern and is usually not applicable while learning from high-dimensional data, due to the large number of missing entries. Moreover, compressed sensing approach usually employs regularizers to control the number of selected features and feature groups. Unfortunately, a clear quantitative relation between the value of regularization and the selected variables/groups, which is often expected in biomedical applications, is hard to establish in most cases.

Motivated by the aforementioned challenges, we carry out research along these directions aiming to develop feature learning models that possess the following char-
acteristics: (1) information fusion from multiple heterogeneous data sources; (2) simultaneous feature and feature group selection; (3) flexibility to handling block-wise missing data without imputation and (4) interpretable model selection. In addition, we also emphasize on investigating and developing efficient optimization algorithms. The rest of this thesis is organized as follows. In Chapter 2, we introduce a unified bi-level learning model that contains several existing feature selection models as special cases. Then in Chapter 3, this proposed model is further extended to tackle the block-wise missing data, one of the major challenges in the diagnosis of Alzheimer’s Disease (AD). Moreover, we propose in Chapter 4 a novel interpretable sparse group feature selection model that greatly facilitates the procedure of parameter tuning and model selection. Last but not least, in Chapter 5, we show that by solving a sparse group hard thresholding problem, the sparse group feature selection model can be further improved in terms of both algorithmic complexity and efficiency. Chapter 6 summarizes the entire thesis, highlight the contributions and points out some promising directions for future work.
Chapter 2

BI-LEVEL LEARNING FOR MULTI-SOURCE COMPLETE DATA

2.1 Introduction

In this chapter, we focus on designing a general bi-level learning model that is capable of performing simultaneous feature-level and source-level learning, assuming that the observed data is complete (See Chapter 3 for extension to handling block-wise missing data). The relationship between our model and existing works and the optimization algorithms are also discussed.

2.2 A Unified Feature Learning Model for Multi-source Data

Assume we are given a collection of $m$ samples from $S$ data sources:

$$X = [X_1, X_2, \cdots, X_S] \in \mathbb{R}^{m \times n}, \quad y \in \mathbb{R}^m,$$

where $X_i \in \mathbb{R}^{m \times p_i}$ is the data matrix of the $i$th source with each sample being a $p_i$-dimensional vector, and $y$ is the corresponding outcome for each sample. We consider the following linear model:

$$y = \sum_{i=1}^{S} X_i \beta_i + \epsilon = X \beta + \epsilon,$$

where each column of $X$ is normalized to be zero mean and standard deviation of 1 and $\epsilon$ represents the noise term. $\beta$ is the underlying true model and is usually unknown in real-world applications. Based on $(X, y)$, we want to learn an estimator of $\beta$, denoted as $\hat{\beta}$, whose non-zero elements $\mathcal{F} = \{j : \hat{\beta}_j \neq 0\}$ correspond to the relevant features. In other words, features correspond to the zero elements of $\hat{\beta}$ are
discarded. We consider the following regularization framework:

$$\min_{\beta} L(\beta) + \Omega(\beta),$$

where $L(\cdot)$ represents the data-fitting term and $\Omega(\cdot)$ is the regularization term which encodes our prior knowledge about $\beta$. Specifically, the choice of $\Omega(\cdot)$ should also enable us to perform both feature-level and source-level analysis simultaneously. Towards this end, a natural approach is a two-stage model. First we learn different models for each data source and then combine these learned models properly. The regularization should be imposed independently on each stage to provide the bi-level analysis. We formalize our intuition as follows:

$$\min_{\alpha, \gamma} \frac{1}{2} \| y - \sum_{i=1}^{S} \gamma_i \cdot X_i \alpha_i \|^2_2 + \sum_{i=1}^{S} \frac{\lambda_i}{p} \| \alpha_i \|^p_p + \sum_{i=1}^{S} \frac{\eta_i}{q} | \gamma_i |^q; \quad (2.2)$$

where the minimization is taken with respect to $(\alpha, \gamma)$ jointly. According to the intuition above, $\alpha_i$ denotes the model learned on the $i$th data source and $\gamma$ is the weight that combines those learned models together. The regularization is taken independently over $\alpha$ and $\gamma$ and therefore we have the flexibility to choose different values of $p$ and $q$ to induce sparsity on either feature-level or source-level. Notice that model (2.2) is not jointly convex and direct optimization towards (2.2) would be difficult. We provide an equivalent but simpler formulation in the following theorem and discuss its optimization in the next section.

**Theorem 1.** The formulation (2.2) is equivalent to the following optimization problem:

$$\min_{\beta} \frac{1}{2} \| y - \sum_{i=1}^{S} X_i \beta_i \|^2_2 + \sum_{i=1}^{S} \nu_i \| \beta_i \|^p_{p+q}. \quad (2.3)$$

**Proof.** Without loss of generality, we assume that $\alpha_i \neq 0$ for all $i = 1, 2, \cdots, S$. Since if $\alpha_i = 0$ for some $i$, the optimal $\gamma_i$ must be 0 and therefore both $\alpha_i$ and $\gamma_i$ can be
removed from (2.2). Let $\beta_i = \gamma_i \cdot \alpha_i$ and replace $\gamma_i$ with $\frac{||\beta_i||_p}{||\alpha_i||_p}$, we can obtain an equivalent formulation:

$$\min_{\alpha, \beta} \frac{1}{2} ||y - \sum_{i=1}^{S} X_i \beta_i||_2^2 + \sum_{i=1}^{S} \frac{\lambda_i}{p} ||\alpha_i||_p^p + \sum_{i=1}^{S} \eta_i \left( \frac{||\beta_i||_p}{||\alpha_i||_p} \right)^q. \quad (2.4)$$

Taking partial derivative with respect to $\alpha_i$ and setting it to zero leads to:

$$\eta_i ||\beta_i||_p^q = \lambda_i ||\alpha_i||_p^{p+q}, \quad i = 1, 2, \ldots, S. \quad (2.5)$$

Plugging (2.3) back into (2.4) with the change of variables, we get the formulation (2.3).

2.2.1 Relation to Previous Works

Formulation (2.2) (or its equivalent form (2.3)) is a very general model. Assigning different values to $p$ and $q$ leads to various kinds of regularization and feature learning models. Next, we show several widely-used convex models are actually our special cases.

Let $p = 1$ and $q = \infty$. In this case, the regularization term in (2.3) becomes the $\ell_1$-regularization and the resulting model becomes Lasso [Tibshirani (1996)]:

$$\min_{\beta} \frac{1}{2} ||y - \sum_{i=1}^{S} X_i \beta_i||_2^2 + \lambda ||\beta||_1. \quad (2.6)$$

It is well-known that the $\ell_1$-regularization leads to a sparse solution, which coincides with the goal of feature selection. However, it does not consider the source structure by treating all features from different sources equally.

On the other hand, if both $p$ and $q$ equal 2, then the $\ell_2$-regularization is applied on each source. Letting $\nu_i = \lambda \sqrt{p_i}$ leads to the group lasso [Yuan and Lin (2006)]:

$$\min_{\beta} \frac{1}{2} ||y - \sum_{i=1}^{S} X_i \beta_i||_2^2 + \lambda \sum_{i=1}^{S} \sqrt{p_i} ||\beta_i||_2. \quad (2.7)$$
Similarly, if $p = \infty$ and $q = 1$, we obtain the $\ell_{1,\infty}$-regularization model \cite{Turlach et al. (2005); Quattoni et al. (2009)}, which penalizes the largest elements of $\beta_i$ for each source:

$$\min_{\beta} \frac{1}{2} \|y - \sum_{i=1}^{S} X_i \beta_i \|_2^2 + \sum_{i=1}^{S} \nu_i \|\beta_i\|_\infty.$$  \hspace{1cm} (2.8)

Besides these common convex formulations, our general model also includes a family of nonconvex formulations which have not been fully explored in the literature. Particularly, letting $p = 1$ and $q = 2$ leads to the following nonconvex model:

$$\min_{\beta} \frac{1}{2} \|y - \sum_{i=1}^{S} X_i \beta_i \|_2^2 + \sum_{i=1}^{S} \nu_i \|\beta_i\|_3.$$  \hspace{1cm} (2.9)

If $p = 2$ and $q = 1$, model (2.3) reduces to:

$$\min_{\beta} \frac{1}{2} \|y - \sum_{i=1}^{S} X_i \beta_i \|_2^2 + \sum_{i=1}^{S} \nu_i \|\beta_i\|_2^3.$$  \hspace{1cm} (2.10)

For the convex models such as lasso, the optimization algorithms have received intensive studies \cite{Barzilai and Borwein (1988); Boyd and Vandenberghe (2004); Efron et al. (2004); Bach (2011)}. In order to fully explore the functionality of our general model, we shall provide further investigations on the nonconvex formulations in terms of optimization.

### 2.3 Optimization

We first focus on formulation (2.10), which is clearly a nonconvex optimization problem. Gasso et al. has shown in \cite{Gasso et al. (2009)} that the $\ell_q$-regularized least squares problem with $q < 1$ can be efficiently solved using the difference of convex functions (DC) algorithm \cite{Tao and An (1997)}. The DC decomposition presented in \cite{Gasso et al. (2009)} requires the regularization term to be a concave function with respect to the absolute value of the variable. However this is not the case in our formulation according to the following proposition:
Proposition 1. Let $f(\beta) = \|\beta\|_2^\frac{3}{2}$. Then $f$ is neither convex nor concave w.r.t. $|\beta|$ unless $\beta$ is a scalar, where $|\cdot|$ denotes the absolute value.

Proof. The proof is carried out by computing the Hessian of $f$. Without loss of generality, we assume $\beta \neq 0$. It can be shown that:
\[
\frac{\partial f}{\partial |\beta_i|} = \frac{2}{3} \|\beta\|_2^{-\frac{4}{3}} |\beta_i|
\]
\[
\frac{\partial^2 f}{\partial |\beta_i| \partial |\beta_j|} = -\frac{8}{9} \|\beta\|_2^{-\frac{10}{3}} |\beta_i \beta_j| + 1_{\{i=j\}} \cdot \frac{2}{3} \|\beta\|_2^{-\frac{4}{3}},
\]
where $1$ is the indicator function. It is clear that, unless $\beta$ is a scalar, in which case it is obvious that $f$ is a concave function, $\frac{\partial^2 f}{\partial |\beta_i| \partial |\beta_i|}$ can be either positive or negative.
In other words, the sign of the diagonal elements of the Hessian of $f$ can be either positive or negative, which means that $f$ is neither convex nor concave.

To employ the DC algorithm, we need to avoid the non-concavity of the regularization item. We introduce new variables $t_i, i = 1, 2, \cdots, S$ and transform (2.9) into the following formulation:
\[
\begin{align*}
&\text{minimize} \quad \frac{1}{2} \|y - \sum_{i=1}^S X_i \beta_i\|_2^2 + \sum_{i=1}^S \nu_i t_i^\frac{2}{3} \\
&\quad \text{subject to} \quad \|\beta_i\|_2 \leq t_i, \quad i = 1, 2, \cdots, S.
\end{align*}
\]
(2.11)
It is clear that (2.11) is equivalent to the original formulation (2.9), however the regularization term in (2.11) is concave with respect to $t_i$, as shown in Proposition 1. We apply the DC algorithm, i.e., for each $t_i^\frac{2}{3}$, we rewrite it as the difference of two convex functions as follows:
\[
t_i^\frac{2}{3} = t_i - (t_i - t_i^\frac{2}{3}).
\]
Therefore, (2.11) becomes:
\[
\begin{align*}
&\text{minimize} \quad \frac{1}{2} \|y - \sum_{i=1}^S X_i \beta_i\|_2^2 + \sum_{i=1}^S \nu_i t_i - \sum_i \nu_i (t_i - t_i^\frac{2}{3}) \\
&\quad \text{subject to} \quad \|\beta_i\|_1 \leq t_i, \quad i = 1, 2, \cdots, S.
\end{align*}
\]
(2.12)
Next we replace the second convex item $t_i - \hat{t}_i^{\frac{2}{3}}$ by its affine minorant at the previous iteration. Specifically, suppose at the previous iteration the value of $t_i$ is $\hat{t}_i$; now we approximate $t_i - \hat{t}_i^{\frac{2}{3}}$ by its first-order Taylor expansion at $\hat{t}_i$ as follows:

$$(\hat{t}_i - \hat{t}_i^{\frac{2}{3}}) + (1 - \frac{2}{3}\hat{t}_i^{-\frac{1}{3}})(t_i - \hat{t}_i).$$

Plugging the above expression back to (2.12) and dropping the constant, we get:

$$\min_{\beta, t} \frac{1}{2}\|y - \sum_{i=1}^{S} X_i \beta_i\|_2^2 + \sum_{i=1}^{S} \frac{2}{3} \hat{t}_i^{-\frac{1}{3}} \nu_i t_i$$

subject to $\|\beta_i\|_2 \leq t_i, \quad i = 1, 2, \ldots, S.$

(2.13)

Since $\nu_i$ and $\hat{t}_i$ are nonnegative, all constraints in (2.13) must be active at the optimal points. Thus, (2.13) is equivalent to the following group lasso problem:

$$\min_{\beta} \frac{1}{2}\|y - \sum_{i=1}^{S} X_i \beta_i\|_2^2 + \sum_{i=1}^{S} \frac{2}{3} \hat{t}_i^{-\frac{1}{3}} \nu_i \|\beta_i\|_2.$$ 

After $\beta$ is obtained, we update $\hat{t}_i$ with $\|\beta_i\|_2$ and continue the iteration until convergence. Notice that $\hat{t}_i^{-\frac{1}{3}}$ can be very large if $\|\beta_i\|_2$ is small. For numerical stability, we add a smoothing term $\theta$ to each $\hat{t}_i$ as suggested by Gasso et al. (2009). The overall procedure is summarized in Algorithm 1.

**Remark 1.** Model (2.13) can be solved in exactly the same way as above. The only difference is in each iteration we need to solve a weighted lasso problem to get $\hat{\beta}(t)$. 

**Remark 2.** Although we only consider the least squares loss function here, the above derivations can be easily extended to other widely-used convex loss functions, such as the logistic function.

2.4 Experiments

To examine the efficacy of the proposed bi-level feature learning models, we report the performance of the proposed models in this section. Specifically, we evaluate the
Algorithm 1 DC algorithm for solving (2.10)

Input: $X, y, \nu$

Output: solution $\beta$ to (2.10)

1: Initialize $\theta, \mu_i(0), i = 1, 2, \cdots, S$

2: for $k = 1, 2, \cdots$ do

3: Update $\beta$ and $\mu_i$ by:

$$\hat{\beta}^k = \arg\min_{\beta \in \mathbb{R}^n} \frac{1}{2} \|y - \sum_{i=1}^{S} X_i \beta_i \|^2_2 + \sum_{i=1}^{S} \mu_i^{k-1} \| \beta_i \|^2_2$$

$$\mu_i^k = \frac{2}{3} \nu_i (\|\hat{\beta}_i^k\|^2_2 + \theta)^{-1/3}, \quad i = 1, 2, \cdots, S.$$ 

4: if the objective stops decreasing then

5: return $\beta = \hat{\beta}^k$

6: end if

7: end for

effectiveness of the complete models (2.9) and (2.10) on synthetic data generated by the linear model (2.1). As the proposed models will be evaluated again in Chapter 5, we omit the results here and refer the readers to Section 5.5, particularly Table 5.3, for details.

2.5 Summary

In this chapter, we propose a unified model for performing bi-level learning on heterogeneous multi-source data. Simultaneous feature selection and feature group selection is enabled by employing proper regularizations. The proposed model also has merit of including common feature learning models as special cases and motivates our investigations in the next few chapters.
Chapter 3

BI-LEVEL MULTI-SOURCE LEARNING WITH HETEROGENEOUS
BLOCK-WISE MISSING DATA FOR ALZHEIMER’S DISEASE PREDICTION

3.1 Introduction

Alzheimers Disease (AD), the most common form of dementia, is a highly prevalent neurodegenerative disease, in which memory and other cognitive functions decline gradually and progressively over time. AD accounts for 50-80% of dementia cases and the number of people affected by AD is expected to increase substantially over the coming decades [Brookmeyer et al. (2007)]. Currently there is no known cure for AD, but the detection and diagnosis of the onset and progression of AD in its earliest stages is invaluable and is the target of intensive investigation world-wide.

Besides the multi-modality and the high dimensionality, the existence of (block-wise) missing data is another major challenge encountered in AD research and other biomedical applications. Figure 3.1 provides an illustration of how block-wise missing data arises in AD research. We have 245 participants in total and 3 types of measurements (PET, MRI and CSF) are taken for diagnosis. Therefore for a single participant, there are at most three different measurements, which are represented in different colors. The blank region means that data from the corresponding source is missing. In this example, participants 1 ~ 60 have records on PET and MRI but lack CSF information while participants 149 ~ 245 have only MRI data. The block-wise missing data situation tends to emerge in several scenarios: low-quality data sources of certain samples may be discarded; some data-collecting mechanisms (like PET) may be too costly to apply to every participant; participants may not be willing to
allow certain measurements, for various reasons (e.g., lack of consent, contraindications, participant attrition, non-compliance with a long scan). Note that the missing data often emerges in a block-wise fashion, i.e., for a patient, a certain data source is either present or missing completely.

![Diagram of incomplete multi-source data with three sources]

**Figure 3.1:** An Illustration of an Incomplete Multi-source Data With Three Sources.

### 3.1.1 Related Work

Considerable efforts have been made to deal with the missing data, both in data mining and biomedical informatics. Some well-known missing value estimation techniques like EM \cite{Duda1997}, iteratively singular value decomposition (SVD) and matrix completion \cite{Mazumder2010} have been extended to biomedical applications by performing imputation on the missing part of the data. Although these approaches have demonstrated their effectiveness on handling random missing entries, they often deliver sub-optimal performance in AD research \cite{Yuan2012} for the following reasons: (1) these imputation approaches fail to capture the pattern of the missing data, i.e., the missing elements are not randomly scattered across the data matrix but emerge block-wisely. However, such prior knowledge is completely discarded in imputation methods; (2) due to the high-dimensionality of the data,
these methods often have to estimate a significant amount of missing values, which would result in unstable performances.

To overcome the aforementioned drawbacks of standard imputation methods, Yuan et al. proposes an incomplete Multi-Source Feature learning method (iMSF) which avoids the direct imputation [Yuan et al. (2012)]. The iMSF method first partitions the patients into disjoint groups such that patients from the same group possess identical data source combinations. Feature learning is then carried out independently in each group and finally the results from all the groups are properly combined to obtain a consistent feature learning result. Such a mechanism enables iMSF to perform feature selection without estimating the missing values, however, the resulting model is unable to provide source-level analysis, i.e., we cannot tell which data source is more important for the diagnosis or which data source should be discarded in a particular application. Such a drawback may limit the performance of iMSF in applications where noisy or corrupted data sources are frequently encountered.

In this chapter, we propose a novel bi-level learning model, which performs simultaneous feature-level and source-level analysis. Bi-level analysis has recently drawn increasing attention [Breheny and Huang (2009); Huang et al. (2012); Xiang et al. (2013a)], but how to extend existing techniques to deal with block-wise missing data remains largely unexplored. We fill in this gap by proposing bi-level feature learning models for block-wise missing data. Our contributions are two-fold: (1) the proposed incomplete model avoids direct imputation of the missing data, and is capable of bi-level feature learning; (2) applying our proposed method incomplete data require solving nonconvex optimization problems. We present efficient optimization algorithms, to find the solution by solving a sequence of convex sub-problems. The proposed incomplete model learns a single model for each data source across different groups (each group corresponds to one data source combination), and learns the
prediction model for each group by computing a weighted combination of the models (one model for each source) involved in the group, thus it provides out-of-sample prediction, overcoming the limitation of the iMSF method. We also evaluate the effectiveness of the proposed models, compared to existing methods using data from the Alzheimer's Disease Neuroimaging Initiative (ADNI). A total of 780 subjects, who have at least one of the four major types of data (MRI, PET, CSF, and proteomics) were available at baseline, and were included in our study. Our experiments show the potential of the proposed models for analyzing multiple heterogeneous sources with block-wise missing data.

3.2 Subjects

We use data from the Alzheimer's disease Neuroimaging Initiative (ADNI) (www.adni-info.org). ADNI was launched in 2003 by the National Institute on Aging (NIA), the National Institute of Biomedical Imaging and Bioengineering (NIBIB), the Food and Drug Administration (FDA), private pharmaceutical companies and non-profit organizations, as a 5-year public private partnership. ADNI's primary goal has been to test whether serial magnetic resonance imaging (MRI), positron emission tomography (PET), other biological markers, and clinical and neuropsychological assessments can be combined to measure the progression of mild cognitive impairment (MCI) and early Alzheimer's disease (AD). Determination of sensitive and specific markers of very early AD progression is intended to aid researchers and clinicians to develop new treatments and monitor their effectiveness, as well as lessen the time and cost of clinical trials. ADNI is the result of efforts of many co-investigators from a broad range of academic institutions and private corporations, and subjects have been recruited from over 50 sites across the U.S. and Canada. ADNI's initial goal was to recruit 800 subjects, but follow-on projects, known as ADNI-GO and ADNI-2, have
recruited over 1500 adults, aged 55 to 90, to participate in the research, consisting of cognitively normal older individuals, people with early or late MCI, and people with early AD. The follow-up intervals for each diagnostic subgroup is specified in the protocols for ADNI-1, ADNI-2 and ADNI-GO. Subjects originally recruited for ADNI-1 and ADNI-GO had the option of being followed longitudinally in ADNI-2. In this work, we use four types of data sources, including MRI, PET, CSF, and proteomics, including a total of 780 subjects (i.e., anyone who had at least one of these measures at baseline). The MRI image features in this study were based on the imaging data from the ADNI database processed by the UCSF team, who performed cortical reconstruction and volumetric segmentations with the FreeSurfer image analysis suite (http://surfer.nmr.mgh.harvard.edu/). We note that many other measures could be, and have been, derived from the MRIs, but this is a representative set, intended to illustrate our approach. The processed MRI features come from a total of 648 subjects (138 AD, 142 progressive MCI, 177 stable MCI and 191 Normal), and may be grouped into 5 categories: average cortical thickness, standard deviation in cortical thickness, the volumes of cortical parcellations, the volumes of specific white matter parcellations, and the total surface area of the cortex. There were 305 MRI features in total. We also downloaded baseline FDG-PET images from 327 subjects (76 AD, 70 progressive MCI, 100 stable MCI and 81 Normal) from the ADNI website. We processed these FDG-PET images using SPM8 (http://www.fil.ion.ucl.ac.uk/spm/). Specifically, we applied Automated Anatomical Labeling (AAL) [Tzourio-Mazoyer et al. (2002)] to extract each of the 116 anatomical volumes of interest (AVOI) and derived average image values from each AVOI, for every subject. Baseline CSF samples were acquired from 409 subjects (100 AD, 84 progressive MCI, 111 stable MCI and 114 Normal) by the ADNI Biomarker Core laboratory at the University of Pennsylvania Medical Center [Tzourio-Mazoyer et al. (2002)]. The proteomics data set (112
AD, 163 progressive MCI, 233 stable MCI and 54 Normal) was produced by the Biomarkers Consortium Project Use of Targeted Multiplex Proteomic Strategies to Identify Plasma-Based Biomarkers in Alzheimer’s Disease. We use 147 measures from the proteomic data downloaded from the ADNI web site. As a result, for a subject with all four types of data available, a total of 571 measures were analyzed in our study. The statistics of these data sources are shown in Table 3.1.

**Table 3.1:** Statistics of the ADNI Data Set and the Data Sources Used In Our Evaluations, Where AD, pMCI, sMCI and NC Stand for Alzheimers Disease Patients, Progressive Mild Cognitive Impairment Patients, Stable Mild Cognitive Impairment Patients, and Normal Controls Respectively.

<table>
<thead>
<tr>
<th></th>
<th>AD</th>
<th>pMCI</th>
<th>sMCI</th>
<th>NC</th>
<th>Samples</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROTEOMICS</td>
<td>112</td>
<td>163</td>
<td>233</td>
<td>58</td>
<td>566</td>
<td>147</td>
</tr>
<tr>
<td>PET</td>
<td>76</td>
<td>70</td>
<td>100</td>
<td>81</td>
<td>327</td>
<td>116</td>
</tr>
<tr>
<td>MRI</td>
<td>138</td>
<td>142</td>
<td>177</td>
<td>191</td>
<td>648</td>
<td>305</td>
</tr>
<tr>
<td>CSF</td>
<td>100</td>
<td>84</td>
<td>111</td>
<td>114</td>
<td>409</td>
<td>3</td>
</tr>
</tbody>
</table>

3.3 Incomplete Source-Feature Selection (iSFS) Model

In this section, we consider the more challenging and more realistic situation with block-wise missing data, as shown in Figure 3.1. In such situation, most patients do not have complete data collected from every data source but lack one or more data blocks. To apply existing feature learning approaches directly, we can either discard all samples that have missing entries or estimate the missing values based on the observed entries. However, the former approach may significantly reduce the size of the data set while the latter approach heavily relies on our prior knowledge about the missing values. Moreover, both approaches neglect the block-wise missing patterns.

---

in the data and therefore could lead to sub-optimal performance.

As in the case of complete data, an ideal model performs both feature-level and source-level analysis simultaneously. Next, we show how to extend the model on complete data presented in the previous section to a more general setting with missing data. Our intuition of designing such Incomplete Source-Feature Selection (iSFS) model is illustrated in Figure 3. We follow a similar strategy used in our complete model (2.2): individual model is learned on each data source and then all models are properly integrated via extra regularizations/constraints. As shown in Figure 3.2, we try to learn the model represented by \( \beta_1, \beta_2, \beta_3 \), corresponding to measurements from PET, MRI and CSF, respectively. A subtle issue is how to learn the coefficients \( \alpha \), since model (2.2) is not applicable due to the presence of missing data blocks. To address this issue, we partition the whole data set into multiple groups according to the availability of data sources, as illustrated in the red boxes in Figure 3.2. For this particular case, we partition the data into 4 groups, where the first group includes all the samples that have PET and MRI, the second group of patients possesses all three data sources, the third group of patients has MRI and CSF measurements, while the last group of patients only has MRI data. Note that within each group we have the complete data and the analysis from the previous section can be applied.

The proposed model is closely related to the iMSF model proposed in Yuan et al. (2012), however, they differ in several significant aspects: (1) the proposed method partitions the data into multiple groups according to the availability of data sources. The resulting groups are not disjoint compared to that of the iMSF. Generally, our partition method results in more samples for each group; (2) in the proposed approach, the model learned for each data source is consistent across different data source combinations while iMSF does not; (3) in every data source combination, we learn the weights of each source from the data. The weights for a specific data source
may differ in different data source combinations. Unlike iMSF, the proposed method achieves source selection by discarding the data sources with a weight of 0. Thus, the proposed method is expected to outperform iMSF especially in the presence of noisy data sources.

![Figure 3.2: Illustration of the Proposed Learning Model.](image)

3.3.1 Formulation

Before presenting the formal description of our iSFS model, we first introduce some notations which will simplify the discussion. Suppose we have $S$ data sources in total and each participant has at least one data source available. Then there are $2^S - 1$ possible missing patterns: the number of all possible combinations of $S$ data sources except for the case that all data sources are missing. For each participant, based on whether a certain data source is present, we obtain a binary indicator vector $I[1 \cdots S]$, where $I[i] = 1$ indicates the $i$th data source is available. For example in Figure 3.1, participants 1 $\sim$ 139 possess the same indicator vector [1, 1, 0] while the indicator vector of participants 149 $\sim$ 245 is [0, 1, 0]. Using such indicator vectors simplifies our analysis. Moreover, we do not even need to store the complete vector.
for each participant but just need to record a single decimal integer if we convert this binary vector to a binary number, i.e., the information in the indicator vector can be completely described by a decimal integer, called profile. All these profiles are stored in an n-dimensional vector pf[1⋯n] where n is the number of participants.

We are ready to give a concise description of our model. Following the aforementioned intuitions, we learn a consistent model (variable \( \beta \)) across different source combinations, while within each combination, the weights (variable \( \alpha \)) for different sources are learned adaptively. Mathematically, the proposed model solves the following formulation:

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{|\text{pf}|} \sum_{m \in \text{pf}} f(X_m, \beta, \alpha_m, y_m) + \lambda R_\beta(\beta) \\
\text{subject to} & \quad R_\alpha(\alpha_m) \leq 1 \quad \forall m \in \text{pf},
\end{align*}
\]  
(3.1)

where

\[
f(X, \beta, \alpha, y) = \frac{1}{n} L \left( \sum_{i=1}^{S} \alpha_i X^i \beta^i, y \right)
\]  
(3.2)

and \( R_\alpha, R_\beta \) are regularizations on \( \alpha, \beta \) respectively. The \( m \) subscript in (3.1) denotes the matrix/vector restricted to the samples that contain \( m \) in their profiles. \( X^i \) and \( \beta^i \) in (3.2) represent the data matrix and and the model of the \( i \)th source, respectively. \( L \) can be any convex loss function such as the least squares loss function or the logistic loss function and \( n \) is number of rows of \( X \).

### 3.3.2 Optimization

One of the advantages of iMSF is its efficient optimization algorithm. In fact, iMSF can be solved by standard convex multi-task learning algorithms [Argyriou et al. (2008); Liu et al. (2009a)]. The proposed iSFS model involves a more complicated optimization problem. In fact, (3.1) is not jointly-convex w.r.t \( \alpha \) and \( \beta \), posing a major challenge. We adapt the alternating minimization method to solve (3.1). More
specifically, we first initialize $\beta$ and compute the optimal $\alpha$. Then $\beta$ is updated based on the computed $\alpha$. We keep this iterative procedure until convergence. For simplicity, we focus on the least squares loss function in the following discussion. The techniques can be easily extended to other loss functions, e.g., the logistic loss function.

**Computing $\alpha$ when $\beta$ is fixed**

As shown in Figure 3.2, we learn the weight $\alpha$ for each source combination independently. Therefore, when $\beta$ is fixed, the objective function of (3.1) is decoupled w.r.t $\alpha_m$ and the optimal $\alpha_m$ is given by the optimal solution of the following problem:

$$\min_{\alpha} \| \sum_{i=1}^{S} \alpha_i X_i \beta_i - y \|_2^2$$

subject to $R(\alpha) \leq 1.$

For many choices of the regularization term $R(\alpha)$, such as the ridge penalty, the $\ell_1$-norm penalty as well as other sparsity-inducing penalties [Bach (2011)], the optimal solution of (3.3) can be efficiently computed via the accelerated gradient algorithm [Beck and Teboulle (2009)].

**Computing $\beta$ when $\alpha$ is fixed**

When we keep $\alpha$ fixed and seek the optimal $\beta$, (3.1) becomes an unconstrained regularization problem:

$$\min_{\beta} g(\beta) + \lambda R(\beta)$$

where

$$g(\beta) = \frac{1}{|\text{pf}|} \sum_{m \in \text{pf}} \frac{1}{2n_m} \| \sum_{i=1}^{S} (\alpha_m X_m^i) \beta_m^i - y_m \|_2^2.$$  

and $n_m$ is number of rows of $X_m$. We can observe that $g(\beta)$ is a quadratic function of $\beta$ and thus the overall formulation is to minimize the summation of a quadratic
term and a regularization term: a typical formulation that can be solved efficiently via accelerated gradient method provided that the following proximal operator Combettes and Pesquet (2010):

\[
\min_{\beta} \frac{1}{2} \| \beta - v \|_2^2 + \lambda R_\beta(\beta)
\]
can be computed efficiently. Indeed, this is the case for many widely used regularization terms. In addition, in order to apply standard first-order lasso solvers, we only need to provide the gradient of \( \beta \) at any given point without knowing the explicit quadratic form. For each data source \( i \), we can compute the gradient of the \( g(\beta) \) w.r.t \( \beta^i \) as follows:

\[
\nabla g(\beta^i) = \frac{1}{|p|} \sum_{m \in p} \frac{1}{n_m} I(m \& 2^{S-i} \neq 0) \left( \alpha_m X_m^i \right)^T \left( \sum_{i=1}^{S} \alpha_m X_m^i \beta_m^i - y_m \right),
\]

where \( I(\cdot) \) is the indicator function which equals 1 when the condition is satisfied and 0 otherwise. The expression \( m \& 2^{S-i} \neq 0 \) ensures that the \( i \)th source exists in the combination \( m \), where \( \& \) denotes the bit-wise AND operation. Then we can obtain \( \nabla g(\beta) \) by stacking all the \( \nabla g(\beta^i), i = 1, 2, \ldots S \) and finally obtain a global solution of (3.4) via applying the accelerated gradient method. Algorithm 2 summarizes our alternating minimization scheme.

Remark 3. Our model can be easily extended to the logistic loss function which is widely used in classification problems. Computing \( \alpha \) in (3.3) amounts to solving a constrained logistic regression problem while computing \( \beta \) in (3.4) requires solving a regularized logistic regression problem. In fact, any convex loss function can be applied to our model as long as the gradient information can be efficiently obtained.

Remark 4. We may apply different forms of \( R_\alpha \) and \( R_\beta \) in order to capture more complex structures, as long as the associated proximal operator can be efficiently
Algorithm 2 Iterative algorithm for solving (3.1)

\textbf{Input:} $X$, $y$, $\lambda$

\textbf{Output:} solution $\alpha$, $\beta$ to (3.1)

1: Initialize $(\beta^0)^k$ by fitting each source individually on the available data.

2: \textbf{for} $k = 1, 2, \cdots$ \textbf{do}

3: \hspace{1em} Compute each $(\alpha)^k$ via solving a constrained lasso problem (3.3).

4: \hspace{1em} Update $(\beta)^k$ via solving a regularized lasso problem (3.4).

5: \hspace{1em} \textbf{if} the objective stops decreasing \textbf{then}

6: \hspace{2em} \textbf{return} $\beta = (\beta)^k$

7: \hspace{1em} \textbf{end if}

8: \hspace{1em} \textbf{end for}

computed. Particularly, we can employ the $\ell_1$-norm penalty to achieve simultaneous feature-level and source-level selection.

Remark 5. A special case of the proposed iSFS model can be obtained by setting $\alpha_m$ to $\frac{1}{n_m}$ for every $m$, where $n_m$ is the number of samples that have profile $m$. As a result, the optimization (3.1) only involves $\beta$ and becomes a convex programming problem. In fact, this is exactly an extension of the classical lasso method to the block-wise missing data. To the best of our knowledge, such an extension is not known in existing literature.

3.4 Experiments

As noted earlier, we utilize the Alzheimer’s Disease Neuroimaging Initiative (ADNI) data set [Mueller et al. (2005); Jack et al. (2008)] and choose 4 data sources for each patient: Proteomics, PET, MRI and CSF. We investigate the classification between AD patient, normal control (NC) subjects, stable MCI subjects (non-converter) and
progressive MCI subjects (converter). Imputation methods such as Mean-value imputation, EM, KNN, iterative SVD and matrix completion as well as the iMSF feature learning model are included for comparison. Notice that kernel learning algorithms are not applicable here since the data are incomplete. All the evaluations are done in a two-stage fashion. In the first stage, we either apply the feature learning methods to select informative features or the imputation methods to fill in the missing entries in the data. Then in the second stage, the Random Forest classifier is applied to perform the classification. We use 10% and 50% of the ADNI data for the training stage respectively and report the accuracy, sensitivity, specificity and the area under the ROC curve (AUC value) on the remaining test data. 5-fold cross-validation is used for selecting suitable parameters for iSFS, iMSF, KNN and SVD. Particularly, for iSFS, iMSF and matrix completion, we choose five values from $[10^{-5}, 10]$ in the log scale as candidates. For KNN, the size of the neighborhood is selected from $[1, 5, 10, 15, 20, 25]$. The rank parameter in the SVD is chosen from $[5, 10, 15, 20, 25, 30]$. In addition, we employ the $\ell_1$-norm penalty for both $R_\alpha$ and $R_\beta$. The results are presented in Table 3.2 to Table 3.7. All the results are averaged over 10 repetitions. From the evaluation results, we can observe that: (1) among all imputation methods, the mean-value imputation and EM demonstrate better performance in terms of accuracy. However, their results are not stable, as revealed by the low sensitivity/specificity value in some tasks; (2) the feature learning models, such as iSFS and iMSF, outperform the imputation methods and often achieve uniform improvement across all the measurements. This coincides with our intuition that estimating the missing blocks directly is usually difficult and unstable and approaches avoiding imputation are preferred. In particular, iSFS clearly delivers the best performance among all approaches. We can also observe from the results that when 10% of the data is used for training, iSFS consistently outperforms iMSF. However, iSFS and iMSF achieve comparable
performance when 50% of the data is used for training. This is consistent with our analysis in Section 4, in which we show that the iSFS formulation can be considered as a constrained version of iMSF and it involves a much smaller number of model parameters than iMSF. Thus, iFSF is expected to outperform iMSF especially when the number of samples in the training set is small.

Table 3.2: Classification Results of AD Patients Versus Normal Controls with 10% Data for Training. All Results Are Averaged Over 10 Replications.

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>iSFS</td>
<td>0.8103</td>
<td>0.8077</td>
<td>0.8124</td>
<td>0.8101</td>
</tr>
<tr>
<td>iMSF</td>
<td>0.7857</td>
<td>0.7671</td>
<td>0.8005</td>
<td>0.7838</td>
</tr>
<tr>
<td>SVD</td>
<td>0.7756</td>
<td>0.7770</td>
<td>0.7746</td>
<td>0.7758</td>
</tr>
<tr>
<td>KNN</td>
<td>0.7668</td>
<td>0.7161</td>
<td>0.8072</td>
<td>0.7617</td>
</tr>
<tr>
<td>Mean</td>
<td>0.7789</td>
<td>0.7845</td>
<td>0.7744</td>
<td>0.7795</td>
</tr>
<tr>
<td>EM</td>
<td>0.8089</td>
<td>0.7963</td>
<td><strong>0.8189</strong></td>
<td>0.8076</td>
</tr>
<tr>
<td>MC</td>
<td>0.5957</td>
<td>0.5710</td>
<td>0.6155</td>
<td>0.5932</td>
</tr>
</tbody>
</table>

3.4.1 Capability of Source Selection

Motivated by the strategies used in Lanckriet et al. (2004), we add two random (noisy) data sources to the ADNI data set to verify the performance of source-level learning. We compare our iSFS model with iMSF and report their performance in Figure 3.3. Besides the previous tasks, two additional evaluations: AD patients vs. MCI and MCI vs. normal controls, are also included. We can see that our method outperforms the iMSF model in most of the cases. Such a result again justifies the importance of source-level analysis when noisy/corrupted data sources are present.
Table 3.3: Classification Results of AD Patients Versus Stable MCI Patients With 10% Data for Training. All Results Are Averaged Over 10 Replications.

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>iSFS</td>
<td>0.7489</td>
<td>0.7032</td>
<td>0.7816</td>
<td>0.7424</td>
</tr>
<tr>
<td>iMSF</td>
<td>0.7172</td>
<td>0.6910</td>
<td>0.7359</td>
<td>0.7135</td>
</tr>
<tr>
<td>SVD</td>
<td>0.6942</td>
<td>0.6510</td>
<td>0.7250</td>
<td>0.6880</td>
</tr>
<tr>
<td>KNN</td>
<td>0.6774</td>
<td>0.6819</td>
<td>0.6742</td>
<td>0.6781</td>
</tr>
<tr>
<td>Mean</td>
<td>0.7338</td>
<td>0.6163</td>
<td><strong>0.8177</strong></td>
<td>0.7170</td>
</tr>
<tr>
<td>EM</td>
<td>0.7174</td>
<td>0.6323</td>
<td>0.7782</td>
<td>0.7052</td>
</tr>
<tr>
<td>MC</td>
<td>0.6234</td>
<td>0.6135</td>
<td>0.6304</td>
<td>0.6220</td>
</tr>
</tbody>
</table>

3.4.2 Benefit of Utilizing Incomplete Data

The proposed approach makes full use of all available data: every sample with at least one available data source could contribute to the overall system. Here we provide a concrete study to show how this could be beneficial and potentially improve the performance. As in the previous evaluations, we utilize the data sources of Proteomics, PET, MRI and CSF, and extract all the samples that have all four data sources. The classification given by iSFS on both complete and incomplete data and other feature learning approaches, including lasso and group lasso (on the smaller complete data) are reported in Figure 3.4, where iSFSC denotes the result given by iSFS on only complete data. We can observe that, by incorporating the information provided by related but incomplete samples, the classification performance on the complete data can be improved substantially.
### Table 3.4: Classification Results of Progressive MCI Patients Versus Normal Controls With 10% Data for Training. All Results Are Averaged Over 10 Replications.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>iSFS</td>
<td>0.8754</td>
<td>0.9361</td>
<td>0.8297</td>
<td>0.8829</td>
</tr>
<tr>
<td>iMSF</td>
<td>0.8611</td>
<td>0.9190</td>
<td>0.8174</td>
<td>0.8682</td>
</tr>
<tr>
<td>SVD</td>
<td>0.7280</td>
<td>0.7222</td>
<td>0.7323</td>
<td>0.7273</td>
</tr>
<tr>
<td>KNN</td>
<td>0.7272</td>
<td>0.6381</td>
<td>0.7944</td>
<td>0.7162</td>
</tr>
<tr>
<td>Mean</td>
<td>0.7889</td>
<td>0.9531</td>
<td>0.6651</td>
<td>0.8091</td>
</tr>
<tr>
<td>EM</td>
<td>0.8027</td>
<td>0.8281</td>
<td>0.7836</td>
<td>0.8059</td>
</tr>
<tr>
<td>MC</td>
<td>0.7740</td>
<td>0.7728</td>
<td>0.7749</td>
<td>0.7738</td>
</tr>
</tbody>
</table>

#### 3.4.3 Ensemble Learning Methods

In this experiment, we employed various ensemble learning approaches to further boost the performance for classification of the ADNI data. Ensemble learning is a commonly used scheme in machine learning and data mining, which properly integrates the models/results learned by different algorithms. In our evaluation, we consider the following two simple ensemble strategies: (1) majority vote; (2) learning the combination coefficients via linear regression. In the first approach, the prediction of a given sample is based on majority voting by all of the algorithms. In other words, all of the participating algorithms are treated equally. By contrast, we learn the combination weights for each algorithm, in the second approach. Therefore the final prediction is based on a weighted-combination of the results obtained from each individual algorithm. Specifically, we include two imputation models: mean-value imputation and KNN. In addition, for each of iMSF and iSFS, we select two parameters (0.001, 0.01), which results in 6 models in total. Figure [here](#) illustrates the
Table 3.5: Classification Results of AD Patients Versus Normal Controls With 50% Data for Training. All Results Are Averaged Over 10 Replications.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>iSFS</td>
<td>0.8848</td>
<td>0.8895</td>
<td>0.8816</td>
<td>0.8856</td>
</tr>
<tr>
<td>iMSF</td>
<td>0.8782</td>
<td>0.8733</td>
<td>0.8816</td>
<td>0.8774</td>
</tr>
<tr>
<td>SVD</td>
<td>0.8469</td>
<td>0.8465</td>
<td>0.8472</td>
<td>0.8469</td>
</tr>
<tr>
<td>KNN</td>
<td>0.8374</td>
<td>0.8407</td>
<td>0.8352</td>
<td>0.8379</td>
</tr>
<tr>
<td>Mean</td>
<td>0.8540</td>
<td>0.8465</td>
<td>0.8592</td>
<td>0.8529</td>
</tr>
<tr>
<td>EM</td>
<td>0.8536</td>
<td>0.8163</td>
<td>0.8792</td>
<td>0.8477</td>
</tr>
<tr>
<td>MC</td>
<td>0.6085</td>
<td>0.5779</td>
<td>0.6296</td>
<td>0.6038</td>
</tr>
</tbody>
</table>

ensemble learning results with varying ratios of training data - we can observe that model ensemble often improves the overall performance of the learning system.

3.4.4 Numerical Results on Algorithm Efficiency

The proposed bi-level learning approach involves solving a nonconvex optimization problem, which is often more difficult than its convex counterpart. Because of the complicated heterogeneity nature of the missing data problem, it is much advantageous to develop an efficient numerical scheme. Our experience shows that the proposed alternating minimization method can achieve a reasonable efficiency performance. Figure 3.6 illustrates the efficiency of Algorithm 1 where the objective value of Eq. 3.1 is plotted as the iteration increases. We can see that the proposed algorithm converges quickly after the first few iterations. We also report the running time of the proposed optimization procedure with increasing number of samples and number of sources in Figure 3.7.
Table 3.6: Classification Results of AD Patients Versus Stable MCI Patients With 50% Data for Training. All Results Are Averaged Over 10 Replications.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>iSFS</td>
<td>0.8603</td>
<td>0.7588</td>
<td>0.9209</td>
<td>0.8384</td>
</tr>
<tr>
<td>iMSF</td>
<td>0.8543</td>
<td>0.7512</td>
<td>0.9142</td>
<td>0.8327</td>
</tr>
<tr>
<td>SVD</td>
<td>0.7808</td>
<td>0.7500</td>
<td>0.7986</td>
<td>0.7743</td>
</tr>
<tr>
<td>KNN</td>
<td>0.7598</td>
<td>0.7570</td>
<td>0.7615</td>
<td>0.7592</td>
</tr>
<tr>
<td>Mean</td>
<td>0.8269</td>
<td>0.6733</td>
<td>0.9162</td>
<td>0.7947</td>
</tr>
<tr>
<td>EM</td>
<td>0.7974</td>
<td>0.7256</td>
<td>0.8392</td>
<td>0.7824</td>
</tr>
<tr>
<td>MC</td>
<td>0.6004</td>
<td>0.6116</td>
<td>0.5939</td>
<td>0.6028</td>
</tr>
</tbody>
</table>

3.5 Summary

In this chapter, we take the prediction of Alzheimer’s Disease as an example and systematically study the bi-level feature learning for block-wise missing data. The proposed model is advantageous in terms of (1) significant reduction of learning parameters by avoiding imputation of missing values; (2) consistent modeling across data sources to ensure out-of-sample capability; (3) source-level selection when noisy/corrupted data sources are present. We also propose efficient numerical schemes to solve the introduced nonconvex optimization problems. Our extensive experiments on ADNI data sets demonstrate the efficacy and efficiency of our proposed framework.
Table 3.7: Classification Results of Progressive MCI Patients Versus Normal Controls With 50% Data for Training. All Results Are Averaged Over 10 Replications.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>iSFS</td>
<td>0.8986</td>
<td><strong>0.9915</strong></td>
<td>0.8400</td>
<td>0.9157</td>
</tr>
<tr>
<td>iMSF</td>
<td><strong>0.9189</strong></td>
<td>0.9622</td>
<td><strong>0.8915</strong></td>
<td><strong>0.9265</strong></td>
</tr>
<tr>
<td>SVD</td>
<td>0.8896</td>
<td>0.9585</td>
<td>0.8462</td>
<td>0.9023</td>
</tr>
<tr>
<td>KNN</td>
<td>0.8288</td>
<td>0.8561</td>
<td>0.8115</td>
<td>0.8338</td>
</tr>
<tr>
<td>Mean</td>
<td>0.6882</td>
<td>0.9976</td>
<td>0.4931</td>
<td>0.7453</td>
</tr>
<tr>
<td>EM</td>
<td>0.8849</td>
<td>0.9902</td>
<td>0.8185</td>
<td>0.9044</td>
</tr>
<tr>
<td>MC</td>
<td>0.7821</td>
<td>0.7829</td>
<td>0.7815</td>
<td>0.7822</td>
</tr>
</tbody>
</table>

Figure 3.3: The Classification Results of iSFS and iMSF on ADNI Data Set With Additional Noisy Data Sources.
Figure 3.4: ROC Curves Given by iSFS (on Both Complete and Incomplete Data), Lasso and Group Lasso.

Figure 3.5: ROC Curves of The Ensemble Methods. The Ratio of The Training Set Varies from 25% to 75% and the Performance On Three Tasks: AD vs. NC, AD vs. Stable MCI and Progressive MCI vs. Normal Controls, Are Reported. The Blue Curve Denotes the Majority Voting Approach, and the Linear Regression Ensemble Method is Represented by the Red Curve.
Figure 3.6: Illustration of the Convergence of Algorithm 1. The X-axis Denotes the Number of Iterations and the Y-axis Denotes the Objective Value of Eq. (3.1).

Figure 3.7: Running Time (in seconds) of the Proposed Algorithm With Increasing Number of Samples and Number of Sources on Synthetic Data.
Chapter 4

INTERPRETABLE BI-LEVEL SELECTION: A CONTINUOUS APPROACH

4.1 Introduction

We have discussed general bi-level selection models for both of complete and block-wise missing data. Remember that in these models, the sparsity, i.e., the number of selected variables and feature groups, is determined by the value of regularizers. One disadvantage of such mechanism is that, we cannot determine the number of selected variables/groups until solving the corresponding optimization problem. In other words, if we want to select 10 features and 3 feature groups, there is no better way but trying different regularizers and running the optimization repeatedly. In this part, we try to facilitate this parameter searching procedure and design new bi-level selection models that incorporate the sparsity in a more interpretable way.

Our work is motivated by the recent advances on nonconvex approximation of the discrete selection problem. It has been shown that nonconvex methods [Fan and Li (2001); Wang et al. (2007); Breheny and Huang (2009); Huang et al. (2009, 2012)], particularly the truncated $L_1$-penalty [Shen et al. (2012); Mazumder et al. (2011); Zhang (2011); Yang et al. (2012b); Sun et al. (2013)], may provide better approximation of the cardinality function and deliver superior performance than the standard $L_1$-formulation. In addition, Shen et al. (2012) suggests that a constrained nonconvex formulation is slightly more preferable than its regularization counterpart due to theoretical merits. In this chapter, we investigate the sparse group feature selection through a constrained nonconvex formulation. Ideally, we wish to optimize the
following $L_0$-model:

$$\begin{align*}
\text{minimize} \quad & \frac{1}{2} \|Ax - y\|_2^2 \\
\text{subject to} \quad & \sum_{j=1}^{p} I(|x_j| \neq 0) \leq s_1 \\
& \sum_{j=1}^{|G|} I(\|x_{G_j}\|_2 \neq 0) \leq s_2,
\end{align*}$$

(4.1)

where $A$ is an $n$ by $p$ data matrix with its columns representing different features. $\mathbf{x} = (x_1, \ldots, x_p)$ is partitioned into $|G|$ non-overlapping groups $\{x_{G_i}\}$ and $I(\cdot)$ is the indicator function. The advantage of the $L_0$-model (4.1) lies in its complete control on two levels of sparsity ($s_1, s_2$), which are the numbers of features and groups respectively. However, problems such like (4.1) are known to be NP-hard because of the discrete nature.

We develop an efficient nonconvex method, which is a computational surrogate of the $L_0$-method described above and has theoretically guaranteed performance. We contribute in two aspects: (i) computationally, we present an efficient optimization algorithm, of which the key step is a projection with two coupled constraints. (ii) statistically, the proposed method retains the merits of the $L_0$ approach in the sense that the oracle estimator can be reconstructed, which leads to consistent feature selection and parameter estimation.

The rest of this chapter is organized as follows. Section 4.2 presents our continuous optimization approach, in which a nonconvex formulation with its optimization algorithm and theoretical properties are explored. The significance of this work is presented in Section 4.3. Section 4.4 demonstrates the efficiency of the proposed methods as well as the performance on real-world applications. Section 4.5 summarizes this chapter.
4.2 Continuous Optimization Approach

One major difficulty of solving (4.1) comes from nonconvex and discrete constraints, which require enumerating all possible combinations of features and groups to achieve the optimal solution. Therefore we approximate these constraints by their continuous computational surrogates:

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \|Ax - y\|_2^2 \\
\text{subject to} & \quad \sum_{j=1}^{p} J_{\tau}(|x_j|) \leq s_1 \\
& \quad \sum_{i=1}^{\mid G \mid} J_{\tau}(\|x_{G_i}\|_2) \leq s_2,
\end{align*}
\]  

(4.2)

where \( J_{\tau}(z) = \min(|z|/\tau, 1) \) is a truncated \( L_1 \)-function approximating the \( L_0 \)-function \cite{Shen2012, Zhang2010}, and \( \tau > 0 \) is a tuning parameter such that \( J_{\tau}(z) \) approximates the indicator function \( I(|z| \neq 0) \) as \( \tau \) approaches zero.

To solve the nonconvex problem (4.2), we develop a Difference of Convex (DC) algorithm \cite{Tao1997} based on a decomposition of each nonconvex constraint function into a difference of two convex functions:

\[
\sum_{j=1}^{p} J_{\tau}(|x_j|) = S_1(x) - S_2(x),
\]

where

\[
S_1(x) = \frac{1}{\tau} \sum_{j=1}^{p} |x_j|, \quad S_2(x) = \frac{1}{\tau} \sum_{j=1}^{p} \max\{|x_j| - \tau, 0\}
\]

are convex in \( x \). Then each trailing convex function, say \( S_2(x) \), is replaced by its affine minorant at the previous iteration

\[
S_1(x) - S_2(\hat{x}^{(m-1)}) - \nabla S_2(\hat{x}^{(m-1)})^T (x - \hat{x}^{(m-1)}),
\]  

(4.3)

which yields an upper approximation of the constraint function \( \sum_{j=1}^{p} J_{\tau}(|x_j|) \) as fol-
lows:

\[
\frac{1}{\tau} \sum_{j=1}^{p} |x_j| \cdot I(|\hat{x}_j^{(m-1)}| \leq \tau) + \sum_{j=1}^{p} I(|\hat{x}_j^{(m-1)}| > \tau) \leq s_1. \tag{4.4}
\]

Similarly, the second nonconvex constraint in (1.2) can be approximated by

\[
\frac{1}{\tau} \sum_{j=1}^{G} \|x_{G_j}\|_2 \cdot I(\|\hat{x}_{G_j}^{(m-1)}\|_2 \leq \tau) + \sum_{j=1}^{G} I(\|\hat{x}_{G_j}^{(m-1)}\|_2 > \tau) \leq s_2. \tag{4.5}
\]

Note that both (4.4) and (4.5) are convex constraints, which result in a convex subproblem as follows:

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \|Ax - y\|_2^2 \\
\text{subject to} & \quad \frac{1}{\tau} \|\hat{x}_{\bar{T_1}(\hat{x}^{(m-1)})}\|_{1} \leq s_1 - (p - |T_1(\hat{x}^{(m-1)})|) \\
& \quad \frac{1}{\tau} \|\hat{x}_{\bar{T_3}(\hat{x}^{(m-1)})}\|_{G} \leq s_2 - (|G| - |T_2(\hat{x}^{(m-1)})|),
\end{align*} \tag{4.6}
\]

where \(T_1, T_2, \text{ and } T_3\) are the support sets \(^1\) defined as:

\[
T_1(x) = \{i : |x_i| \leq \tau\}, \quad T_2(x) = \{i : \|x_{G_i}\|_2 \leq \tau\}
\]

\[
T_3(x) = \{i : x_i \in x_{G_j}, j \in T_2(x)\},
\]

\(\|x_{T_1}\|_1\) and \(\|x_{T_3}\|_G\) denote the corresponding value restricted on \(T_1\) and \(T_3\) respectively, and \(\|x\|_G = \sum_{i=1}^{G} \|x_{G_i}\|_2\). Solving (4.6) would provide us an updated solution, denoted as \(\hat{x}^{(m)}\), which leads to a refined formulation of (1.3). Such procedure is iterated until the objective value stops decreasing. The DC algorithm is summarized in Algorithm 3, from which we can see that efficient computation of (4.6) is critical to the overall DC routine. We defer detailed discussion of this part to Section 4.2.1.

### 4.2.1 Optimization Procedures

As mentioned in our previous discussion, efficient computation of the convex subproblem (4.6) is of critical importance for the proposed DC algorithm. Note that (4.6)\(^1\)

---

\(^1\)Support sets indicate that the elements outside these sets have no effect on the particular items in the constraints of (1.3).
Algorithm 3 DC programming for solving (1.2)

Input: $A$, $y$, $s_1$, $s_2$

Output: solution $x$ to (1.2)

1: Initialize $\hat{x}^{(0)}$.
2: for $m = 1, 2, \cdots$ do
3: Compute $\hat{x}^{(m)}$ by optimizing (1.3).
4: Update $T_1$, $T_2$ and $T_3$.
5: if the objective stops decreasing then
6: return $x = \hat{x}^{(m)}$
7: end if
8: end for

has an identical form of the constrained sparse group lasso problem:

$$
\begin{align*}
\min_{x} & \quad \frac{1}{2}\|Ax - y\|_2^2 \\
\text{subject to} & \quad \|x\|_1 \leq s_1 \\
& \quad \|x\|_G \leq s_2
\end{align*}
$$

(4.7)

except that $x$ is restricted to the two support sets. As to be shown in Section 4.2.1, an algorithm for solving (1.0) can be obtained through only a few modifications on that of (1.7). Therefore, we first focus on solving (1.7). Notice that if problem (1.7) has only one constraint, the solution is well-established [Duchi et al. (2008); Bach et al. (2011)]. However, the two coupled constraints here make the optimization problem more challenging to vvsolve.

Accelerated Gradient Method

For large-scale problems, the dimensionality of data can be very high, therefore first-order optimization is often preferred. We adapt the well-known accelerated gradient
method (AGM) \cite{Nesterov2007}; \cite{Beck}, which is commonly used due to its fast convergence rate.

To apply AGM to our formulation (4.7), the crucial step is to solve the following Sparse Group Lasso Projection (SGLP):

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2}\| x - v \|^2_2 \\
\text{subject to} & \quad \| x \|_1 \leq s_1 \quad (C_1) \\
& \quad \| x \|_G \leq s_2 \quad (C_2),
\end{align*}
\]

which is an Euclidean projection onto a convex set and a special case of (4.7) when \( A \) is the identity. For convenience, let \( C_1 \) and \( C_2 \) denote the above two constraints in what follows.

Since the AGM is a standard framework whose efficiency mainly depends on that of the projection step, we leave the detailed description of AGM in the Appendix and introduce the efficient algorithm for this projection step (4.8).

Efficient Projection

We begin with some special cases of (4.8). If only \( C_1 \) exists, (4.8) becomes the well-known \( L_1 \)-ball projection \cite{Duchi}, whose optimal solution is denoted as \( \mathcal{P}_{s_1}(v) \), standing for the projection of \( v \) onto the \( L_1 \)-ball with radius \( s_1 \). On the other hand, if only \( C_2 \) is involved, it becomes the group lasso projection, denoted as \( \mathcal{P}_{s_2}^G \). Moreover, we say a constraint is active, if and only if an equality holds at the optimal solution \( x^* \); otherwise, it is inactive.

Preliminary results are summarized in Lemma 1:

**Lemma 1.** Denote a global minimizer of (4.8) as \( x^* \). Then the following results hold:

1. If both \( C_1 \) and \( C_2 \) are inactive, then \( x^* = v \).
2. If $C_1$ is the only active constraint, i.e., $\|x^*\|_1 = s_1$, $\|x^*\|_G < s_2$, then $x^* = P_{1}^{s_1}(v)$

3. If $C_2$ is the only active constraint, i.e., $\|x^*\|_1 < s_1$, $\|x^*\|_G = s_2$, then $x^* = P_{G}^{s_2}(v)$

**Computing $x^*$ from the optimal dual variables:** Lemma 1 describes a global minimizer when either constraint is inactive. Next we consider the case in which both $C_1$ and $C_2$ are active. By the convex duality theory [Boyd and Vandenberghe (2004)], there exist unique non-negative dual variables $\lambda^*$ and $\eta^*$ such that $x^*$ is also the global minimizer of the following regularized problem:

$$
\text{minimize} \quad \frac{1}{2}\|x - v\|_2^2 + \lambda^*\|x\|_1 + \eta^*\|x\|_G; \tag{4.9}
$$

whose solution is given by the following Theorem.

**Theorem 2** ([Friedman et al. (2010)]). The optimal solution $x^*$ of (4.9) is given by

$$(4.10)$$

$$
x_{G_i}^* = \max\{\|v_{G_i}^\lambda\|_2 - \eta^*, 0\} \frac{v_{G_i}^\lambda}{\|v_{G_i}^\lambda\|_2} \quad i = 1, 2, \ldots, |G|
$$

where $v_{G_i}^\lambda$ is computed via soft-thresholding [Donoho (2002)] $v_{G_i}$ with threshold $\lambda^*$ as follows:

$$
v_{G_i}^\lambda = SGN(v_{G_i}) \cdot \max\{|v_{G_i}| - \lambda^*, 0\},
$$

where $SGN(\cdot)$ is the sign function and all the operations are taken element-wise.

Theorem 2 gives an analytical solution of $x^*$ in an ideal situation when the values of $\lambda^*$ and $\eta^*$ are given. Unfortunately, this is not the case and the values of $\lambda^*$ and $\eta^*$ need to be computed directly from (4.8). Based on Theorem 2, we have the following conclusion characterizing the relations between the dual variables:
Corollary 1. The following equations hold:

\[ \|x^*\|_1 = \sum_{i=1}^{[G]} \max\{\|v_{G_i}^{\lambda^*}\|_2 - \eta^*, 0\} \frac{\|v_{G_i}^{\lambda^*}\|_1}{\|v_{G_i}^{\lambda^*}\|_2} = s_1 \quad (4.11) \]

\[ \|x^*\|_G = \sum_{i=1}^{[G]} \max\{\|v_{G_i}^{\lambda^*}\|_2 - \eta^*, 0\} = s_2 \quad . \quad (4.12) \]

Suppose \( \lambda^* \) is given, then computing \( \eta^* \) from \((4.12)\) amounts to solving a median finding problem, which can be done in linear time [Duchi et al. (2008)].

Finally, we treat the case of unknown \( \lambda^* \) (thus unknown \( \eta^* \)). We propose an efficient bisection approach to compute it.

**Computing \( \lambda^* \) via bisection:** Given an initial guess (estimator) of \( \lambda^* \), say \( \hat{\lambda} \), one may perform bisection to locate the optimal \( \lambda^* \), provided that there exists an oracle procedure indicating if the optimal value is greater than \( \hat{\lambda} \). This bisection method can estimate \( \lambda^* \) in logarithm time. Next, we shall design an oracle procedure.

Let the triples

\( (x^*, \lambda^*, \eta^*) = \text{SGLP}(v, s_1, s_2) \)

be the optimal solution of \((4.8)\) with both constraints active, i.e., \( \|x^*\|_1 = s_1, \|x^*\|_G = s_2 \), with \( (\lambda^*, \eta^*) \) be the optimal dual variables. Consider the following two sparse group lasso projections:

\( (x, \lambda, \eta) = \text{SGLP}(v, s_1, s_2), \)

\( (x', \lambda', \eta') = \text{SGLP}(v, s_1', s_2'). \)

The following key result holds.

**Theorem 3.** If \( \lambda \leq \lambda' \) and \( s_2 = s_2' \), then \( s_1 \geq s_1' \).

We give the proof of Theorem 3 by preseting a more general conclusion.

\(^2\)An upper bound and a lower bound of \( \lambda^* \) should be provided in order to perform the bisection. These bounds can be easily derived from the assumption that both \( C_1 \) and \( C_2 \) are active.
Lemma 2. Let $\Omega \subset \mathbb{R}^n$ and consider the following optimization problem:

$$\min_{x \in \Omega} F(x) = f(x) + \lambda g(x),$$

where $g$ is non-negative. Let $x_1$ and $x_2$ be the optimal solution of this optimization problem with $\lambda = \lambda_1$ and $\lambda = \lambda_2$ respectively and suppose $\lambda_1 < \lambda_2$. Then we have:

1. $g(x_1) \geq g(x_2)$
2. $F(x_1) \leq F(x_2)$
3. $f(x_1) \leq f(x_2)$

Proof. From the optimality of $x_1$ and $x_2$ we can obtain:

$$f(x_1) + \lambda_1 g(x_1) \leq f(x_2) + \lambda_1 g(x_2)$$
$$f(x_2) + \lambda_2 g(x_2) \leq f(x_1) + \lambda_2 g(x_1).$$

Add the above two equalities leads to $(\lambda_1 - \lambda_2)(g(x_1) - g(x_2)) \leq 0$ which proves the first result. The second conclusion can be shown in the following relation, where we use the non-negativity of $g$ in the second inequality:

$$F(x_1) = f(x_1) + \lambda_1 g(x_1) \leq f(x_2) + \lambda_1 g(x_2) \leq f(x_2) + \lambda_2 g(x_2) = F(x_2).$$

The third conclusion follows directly from the first two results. \hfill \Box

Proof of Theorem 3. Let $\Omega = \{ x : \|x\|_G = s_2 \}$, $g(x) = \|x\|_1$ and applying Lemma 2 gives the result. \hfill \Box

Theorem 3 gives exactly the oracle procedure we need. For a given estimator $\hat{\lambda}$, we compute its corresponding $\hat{\eta}$ from (4.12) and then $\hat{s}_1$ from (4.11), satisfying $(\hat{x}, \hat{\lambda}, \hat{\eta}) = \text{SGLP}(v, \hat{s}_1, s_2)$. Then $\hat{s}_1$ is compared with $s_1$. Clearly, by Theorem 3, if $\hat{s}_1 \leq s_1$, the estimator $\hat{\lambda}$ is no less than $\lambda^*$. Otherwise, $\hat{s}_1 > s_1$ means $\hat{\lambda} < \lambda^*$. 40
In addition, from (4.11) we know that $s_1$ is a continuous function of $\hat{\lambda}$. Together with the monotonicity given in Theorem 3, a bisection approach can be employed to calculate $\lambda^*$. Algorithm 4 gives a detailed description of this bisection procedure and Algorithm 5 summarizes the entire projection method.

Algorithm 4 The Bisection Procedure

Function $\text{bisection}(\mathbf{v}, s_1, s_2)$

1: Initialize $\text{up}$, $\text{low}$ and $\text{tol}$

2: while $\text{up} - \text{low} > \text{tol}$ do

3: $\hat{\lambda} = (\text{low} + \text{up})/2$

4: if (4.12) has a solution $\hat{\eta}$ given $\mathbf{v}^{\hat{\lambda}}$ then

5: calculate $\hat{s}_1$ using $\hat{\eta}$ and $\hat{\lambda}$.

6: if $\hat{s}_1 \leq s_1$ then

7: $\text{up} = \hat{\lambda}$

8: else

9: $\text{low} = \hat{\lambda}$

10: end if

11: else

12: $\text{up} = \hat{\lambda}$

13: end if

14: end while

15: $\lambda^* = \text{up}$

16: Solve (4.12) to get $\eta^*$

17: Calculate $\mathbf{x}^*$ from $\lambda^*$ and $\eta^*$ via (4.10)

18: return $\mathbf{x}^*$

Remark 1. In Su et al.’s work [Su et al. (2012)], the authors develop similar bisection algorithm for solving Problem (4.8). However, both works are accomplished
Algorithm 5 Sparse Group Lasso Projection Algorithm

**Input:** $v, s_1, s_2$

**Output:** an optimal solution $x$ to the Sparse Group Projection Problem

**Function** SGLP($v, s_1, s_2$)

1: if $\|x\|_1 \leq s_1$ and $\|x\|_G \leq s_2$ then
2: return $v$
3: end if
4: $x_{C_1} = P_{s_1}^1(v)$
5: $x_{C_2} = P_{s_2}^G(v)$
6: $x_{C_{12}} = \text{bisec}(v, s_1, s_2)$
7: if $\|x_{C_1}\|_G \leq s_2$ then
8: return $x_{C_1}$
9: else if $\|x_{C_2}\|_1 \leq s_1$ then
10: return $x_{C_2}$
11: else
12: return $x_{C_{12}}$
13: end if

independently and are publicly available around the same time.

Solving Restricted version of (1.7)

Finally, we modify the above procedures to compute the optimal solution of the restricted problem (1.5). To apply the accelerated gradient method, we consider the
following projection step:

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \|x - v\|_2^2 \\
\text{subject to} & \quad \|x^T_1\|_1 \leq s_1 \quad (C_1) \\
& \quad \|x^T_3\|_G \leq s_2 \quad (C_2).
\end{align*}
\]

Our first observation is: \(T_3(x) \subset T_1(x)\), since if an element of \(x\) lies in a group whose \(L_2\)-norm is less than \(\tau\), then the absolute value of this element must also be less than \(\tau\). Secondly, from the decomposable nature of the objective function, we conclude that:

\[
x_j^* = \begin{cases} 
  v_j & \text{if } j \in (T_1)^c \\
  v_j^{\lambda^*} & \text{if } j \in T_1 \setminus T_3,
\end{cases}
\]

since there are no constraints on \(x_j\) if it is outside \(T_1\) and involves only the \(L_1\)-norm constraint if \(j \in T_1 \setminus T_3\). Following routine calculations as in [Duchi et al. (2008)], we obtain the following results similar to (4.11) and (4.12):

\[
\begin{align*}
  s_1 &= \sum_{i \in T_2} \max\{\|v_G^\lambda\|_2 - \eta^*, 0\} \frac{\|v_G^\lambda\|_1}{\|v_G^\lambda\|_2} + \sum_{j \in T_1 \setminus T_3} v_j^{\lambda^*} \\
  s_2 &= \sum_{i \in T_2} \max\{\|v_G^\lambda\|_2 - \eta^*, 0\}.
\end{align*}
\]

Based on (4.14) and (4.15), we design a similar bisection approach to compute \(\lambda^*\) and thus \((x^*)^T_3\), as in Algorithm 5. Details can be found in the Appendix.

Since the projection (4.13) does not possess an closed-form, it is instructive to discuss the convergence property of overall accelerated gradient method. Follow the discussion in [Schmidt et al. (2011)], we can provide sufficient conditions for a guaranteed convergence rate. Moreover, we found in practice that a reasonable convergence property can be obtained as long as the precision level for the computation of the projection is small, as revealed in Section 4.3.

**Remark 2.** Problem (4.7) can also be solved using the Alternating Direction Method of Multiplier (ADMM) [Boyd et al. (2011)] instead of the accelerated gradient method.
(AGM). However, our evaluations show that AGM with our projection algorithm is more efficient than ADMM.

4.2.2 Theoretical Results

This section investigates theoretical aspects of the proposed method. More specifically, we show that the oracle estimator $\hat{x}^o$, the least squares estimator based on the true model, can be reconstructed by a global minimizer of (2). As a result, consistent selection as well as optimal parameter estimation can be achieved by our method.

For presentation, we introduce some notations to be used subsequently. Let $G = (G_{i_1}, \cdots, G_{i_k})$ be a collection of groups, each containing nonzero elements. Let $B_{G_j} = B_{G_j}(x)$ and $B_G = B_G(x)$ denote the indices of nonzero elements of $x$ in $G_j$ and $x$, respectively. Define

$$S_{j,i} = \{x \in S : B_G \neq B_{G^0}, |B_G| = j, |G| = i\},$$

where $S$ is a feasible region of (4.2) and $G^0$ represents the true nonzero groups.

Let $G^0$ and $x^0$ denote the true nonzero groups and the true parameter under $G^0$. The following conditions are assumed to establish consistent reconstruction of the oracle estimator:

**Assumption 1** (Degree of group separation). Define

$$C_{\min}(x^0) = \inf_{x \in S : B_G \neq B_{G^0}} \frac{-\log(1 - h^2(x, x^0))}{\max(|B_{G^0} \setminus B_G|, 1)},$$

then for some constant $c_1 > 0$,

$$C_{\min}(x^0) \geq c_1 \frac{\log |G| + \log s^0_1}{n},$$

where

$$h(x, x^0) = \left(\frac{1}{2} \int (g^{1/2}(x, y) - g^{1/2}(x^0, y))^2 d\mu(y)\right)^{1/2}$$

is the Hellinger-distance for densities with respect to a dominating measure $\mu$. 

44
Assumption 2 (Complexity of the parameter space). For some constants \(c_0 > 0\) and any \(0 < t < \varepsilon \leq 1\),

\[
H(t, \mathcal{F}_{j,i}) \leq c_0 \max(\log(|G| + s_1^0)^2, 1)|B_{j,i}| \log(2\varepsilon/t),
\]

where \(B_{j,i} = S_{j,i} \cap \{x \in h(x, x^0) \leq 2\varepsilon\}\) is a local parameter space and \(\mathcal{F}_{j,i} = \{g^{1/2}(x, y) : x \in B_{j,i}\}\) is a collection of square-root densities. \(H(\cdot, \mathcal{F})\) is the bracketing Hellinger metric entropy of space \(\mathcal{F}\) [Kolmogorov and Tihomirov (1961)].

Assumption 3. For some positive constants \(d_1, d_2, d_3\) with \(d_1 > 10\),

\[
- \log(1 - h^2(x, x^0)) \geq -d_1 \log(1 - h^2(x^\tau, x^0)) - d_3 \tau^{d_3} p,
\]

where \(x^\tau = (x_1 I(|x_1| \geq \tau), \ldots, x_p I(|x_p| \geq \tau)).\)

Under these assumptions, we derive a non-asymptotic error bound regarding the reconstruction of the oracle estimator \(\hat{x}^o\). The proof is provided in the Appendix.

Theorem 4. Suppose that Assumptions 2 and 3 hold. For a global minimizer of \(\hat{x}\) with \((s_1, s_2) = (s_1^0, s_2^0)\) and \(\tau \leq \left(\frac{(d_1-10)C_{\min}(x^0)}{d_3d}\right)^{1/d_3}\), the following result holds:

\[
P\left(\hat{x} \neq \hat{x}^o\right) \leq \exp\left(-c_2 n C_{\min}(x^0) + 2(\log |G| + \log s_1^0)\right).
\]

Moreover, under Assumption 2, \(P\left(\hat{x} = \hat{x}^o\right) \to 1\) and

\[
E h^2(\hat{x}, x^o) = (1 + o(1)) \max(E h^2(\hat{x}^o, x^0), \frac{s_1^0}{n})
\]

as \(n \to \infty, |G| \to \infty\).

Theorem 4 says that the oracle estimator \(\hat{x}^o\) can be accurately reconstructed, which in turn yields feature selection consistency as well as the recovery of the optimal performance of the oracle estimator in the Hellinger distance in (4.16). Moreover, as indicated in 2, the asymptotic result in Theorem 4 holds when \(s_1^0|G|\) grows in the
order of $\exp(c^{-1}_1 n C_{\min})$. This is in contrast to existing results on consistent feature selection, where the number of candidate features should be no greater than $\exp(c^*_n)$ for some $c^*$ [Zhao and Yu (2006); Wang et al. (2007)]. In this sense, the number of candidate features is allowed to be much larger when an additional group structure is assumed, particularly when each group contains a large number of redundant features. It remains unclear whether such a result continues to hold for other bi-level variable selection methods, such as the composite MCP [Huang et al. (2009)] and group bridge [Breheny and Huang (2009)].

To our knowledge, our theory for the grouped selection is the first of this kind. However, it has a root in feature selection. The large deviation approach used here is applicable to derive bounds for feature selection consistency. In such a situation, the result agrees with the necessary condition for feature selection consistency for any method, except for the constants independent of the sample size [Shen et al. (2012)]. In other words, the required conditions are weaker than those for $L_1$-regularization commonly used in the literature [Van De Geer and Buhlmann (2009)]. The use of the Hellinger-distance is to avoid specifying a sub-Gaussian tail of the random error. This means that the result continues to hold even when the error does not have a sub-Gaussian tail. This is because of the one-sided property of the likelihood ratios [Wong and Shen (1995)].

Remark 3. Although we require $\hat{x}$ to be a global minimizer of (1.2), a weaker version of the theory can be derived for a local minimizer obtained from the DC programming by following similar derivations in [Shen et al. (2013)], and will not pursue this direction in here.

Now we consider a special case that the random error follows a gaussian distribution. Specifically assume the response vector $y$ of $n$ observations follows a linear
model as follows:

\[ y = Ax + \varepsilon, \]  

(4.17)

where the random vector \( \varepsilon \) follows \( N(0, \sigma^2 I) \). We can obtain a simplified version of Theorem 4 where both of the \( L_2 \)-norm and Hellinger distance can be applied.

**Proposition 2.** Under the linear model (4.17), suppose \( x \) is uniformly bounded away from infinity and

\[ \gamma_{\min}^2 \min_{B:|B|\leq 2|B_{c0}|,B_{c0}\subset B} c_{\min}(\Sigma_B) \geq c_1 \frac{\log |G| + \log s_0}{n}, \]

for some constant \( c_1 > 0 \), where \( \gamma_{\min} \) is the smallest absolute nonzero element of \( x^0 \), and \( \Sigma_B \) is the covariance matrix indexed by subset \( B \), and \( c_{\min} \) is a minimum eigenvalue of a matrix. Then all the results in Theorem 4 hold for both of the Hellinger distance and the \( L_2 \)-norm.

### 4.3 Discussions

This section is devoted to a brief discussion of advantages of our work statistically and computationally. Moreover, it explains why the proposed methods is useful to perform efficient and interpretable feature selection given a natural group structure.

**Interpretability.** The parameters in the proposed method are highly interpretable in that \( s_1 \) and \( s_2 \) are upper bounds of the number of nonzero elements as well as that of groups. This is advantageous, especially in the presence of certain prior knowledge regarding the number of features and/or that of groups. However, such an interpretation vanishes with other (convex & nonconvex) methods such as lasso, sparse group lasso, composite MCP or group bridge, in which incorporating such prior knowledge often requires repeated trials of different parameters.

**Parameter tuning.** Typically, tuning parameters for good generalization usually requires considerable amount work due to a large number of choices of parameters.
However, parameter tuning in model (4.1) may search through integer values in a bounded range, and can be further simplified when certain prior knowledge is available. This permits more efficient tuning than its regularization counterpart. Based on our limited experience, we note that $\tau$ does not need to be tuned precisely as we may fix at some small values.

**Performance and Computation.** Although our model (4.2) is proposed as a computational surrogate of the ideal $L_0$-method, its performance can also be theoretically guaranteed, i.e., consistent feature selection can be achieved. Moreover, the computation of our model is much more efficient and applicable to large-scale applications.

4.4 Experiments

4.4.1 Evaluation of Projection Algorithms

Since DC programming and the accelerated gradient methods are both standard, the efficiency of the proposed nonconvex formulation (4.2) depends on the projection step in (4.8). Therefore, we focus on evaluating the projection algorithms and comparing with two popular projection algorithms: Alternating Direction Method of Multiplier (ADMM) [B Boyd et al. (2011)] and Dykstra’s projection algorithm [Combettes and Pesquet (2010)]. We give a detailed derivation of adapting these two algorithms to our formulation in the Appendix.

To evaluate the efficiency, we first generate the vector $\mathbf{v}$ whose entries are uniformly distributed in $[-50, 50]$ and the dimension of $\mathbf{v}$, denoted as $p$, is chosen from the set \{10^2, 10^3, 10^4, 10^5, 10^6\}. Next we partition the vector into 10 groups of equal size. Finally, $s_2$ is set to $5\log(p)$ and $s_1$, the radius of the $L_1$-ball, is computed by $\frac{\sqrt{10}}{2}s_2$ (motivated by the fact that $s_1 \leq \sqrt{10}s_2$).

For a fair comparison, we run our projection algorithm until converge and record
the minimal objective value as \( f^* \). Then we run ADMM and Dykstra’s algorithm until their objective values become close to ours. More specifically, we terminate their iterations as soon as \( f_{\text{ADMM}} - f^* \leq 10^{-3} \) and \( f_{\text{Dykstra}} - f^* \leq 10^{-3} \), where \( f_{\text{ADMM}} \) and \( f_{\text{Dykstra}} \) stand for the objective value of ADMM and Dykstra’s algorithm respectively. Table 4.1 summarizes the average running time of all three algorithms over 100 replications.

**Table 4.1:** Running Time (in seconds) of Dykstra’s, ADMM and Our Projection Algorithm. All Three Algorithms Are Averaged Over 100 Replications.

<table>
<thead>
<tr>
<th>Methods</th>
<th>(10^2)</th>
<th>(10^3)</th>
<th>(10^4)</th>
<th>(10^5)</th>
<th>(10^6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dykstra</td>
<td>0.1944</td>
<td>0.5894</td>
<td>4.8702</td>
<td>51.756</td>
<td>642.60</td>
</tr>
<tr>
<td>ADMM</td>
<td>0.0519</td>
<td>0.1098</td>
<td>1.2000</td>
<td>26.240</td>
<td>633.00</td>
</tr>
<tr>
<td>Ours</td>
<td>&lt; (10^{-7})</td>
<td>0.0002</td>
<td>0.0051</td>
<td>0.0440</td>
<td>0.5827</td>
</tr>
</tbody>
</table>

Next we demonstrate the accuracy of our projection algorithm. Toward this end, the general convex optimization toolbox CVX [Grant and Boyd (2011)] is chosen as the baseline. Following the same strategy of generating data, we report the distance (computed from the Euclidean norm \( \| \cdot \|_2 \)) between optimal solution of the three projection algorithms and that of the CVX as well as the running time. Note that the projection is strictly convex with a unique global optimal solution.

For ADMM and Dykstra’s algorithm, the termination criterion is that the relative difference of the objective values between consecutive iterations is less than a threshold value. Specifically, we terminate the iteration if \( |f(x_{k-1}) - f(x_k)| \leq 10^{-7}f(x_{k-1}) \). For our projection algorithm, we set the \( \text{tol} \) in Algorithm 5 to be \( 10^{-7} \). The results are summarized in Table 4.2 and Figure 4.1. Powered by second-order optimization algorithms, CVX can provide fast and accurate solutions for medium-size problems but would suffer from great computational burden for large-scale ones. Therefore we
only report the results up to 5,000 dimensions.

Table 4.2: Distance Between the Optimal Solution of Projection Algorithms and That of The CVX. All The Results Are Averaged Over 100 Replications.

<table>
<thead>
<tr>
<th>Methods</th>
<th>50</th>
<th>100</th>
<th>500</th>
<th>1000</th>
<th>5000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dykstra</td>
<td>9.00</td>
<td>9.81</td>
<td>11.40</td>
<td>11.90</td>
<td>12.42</td>
</tr>
<tr>
<td>ADMM</td>
<td>0.64</td>
<td>0.08</td>
<td>3.6e-3</td>
<td>6.3e-3</td>
<td>1.3e-2</td>
</tr>
<tr>
<td>ours</td>
<td>1.4e-3</td>
<td>1.1e-3</td>
<td>1.2e-3</td>
<td>1.7e-3</td>
<td>7.3e-3</td>
</tr>
</tbody>
</table>

Figure 4.1: The Average Running Time for Different Algorithms To Achieve the Precision Level Listed in Table 4.2.

From the above results we can observe that for projections of a moderate size, all three algorithms perform well. However, for large-scale ones, the advantage of the proposed algorithm is evident as our method provides more accurate solution with less time.
4.4.2 Performance on Synthetic Data

We generate a $60 \times 100$ matrix $A$, whose entries follow i.i.d standard normal distribution. The 100 features (columns) are partitioned into 10 groups of equal size. The ground truth vector $x_0$ possesses nonzero elements only in 4 of the 10 groups. In addition, only 4 elements in each nonzero group are nonzero. Finally $y$ is generated according to $Ax_0 + z$ with $z$ following distribution $N(0, 0.5^2)$. The data are divided into training and testing set of equal size.

We fit our methods to the training set and compare with both convex methods (lasso, group lasso and sparse group lasso) and methods based on nonconvex bi-level penalties (group bridge and composite MCP). Since the data are intentionally generated to be sparse in both group-level and feature-level, approaches that only perform group selection, such as group lasso, group SCAD and ordinary group MCP, are not included due to their suboptimal results.

The tuning parameters of the convex methods are selected from the following set $\{0.01, 0.1, 1, 10\}$, whereas for our methods, the number of nonzero groups ($s_2$) is selected from the set $\{2, 4, 6, 8\}$ and the number of features ($s_1$) is chosen from the set $\{2s_2, 4s_2, 6s_2, 8s_2\}$. 10-fold cross validation is taken for parameter tuning. Group bridge and composite MCP are carried out using their original R-package grpreg and the tuning parameters are set to the default values (100 parameters with 10-fold cross-validation).

Following similar settings in Breheny and Huang (2009), we list the number of selected groups and features by each method. In addition, the number of false positive or false negative groups/features are also reported in Table 4.3. We can observe that our models correctly identify the underlying groups and features. Moreover, our methods effectively exclude redundant features and groups compared to other meth-

51
ods, which is illustrated by our low false positive numbers and relatively high false negative numbers. Such a phenomenon also appears in the evaluations in [Breheny and Huang (2009)].

Table 4.3: Comparison of Performance on Synthetic Data. All the Results Are Averaged for 100 Replications. DC, Which Stands for Difference of Convex functions, Denotes Our Proposed Method.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Groups</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NO.1</td>
<td>FP²</td>
</tr>
<tr>
<td>LASSO</td>
<td>7.56</td>
<td>3.85</td>
</tr>
<tr>
<td>SGL</td>
<td>7.29</td>
<td>3.68</td>
</tr>
<tr>
<td>DC</td>
<td>3.37</td>
<td>0.81</td>
</tr>
<tr>
<td>cMCP</td>
<td>9.5</td>
<td>5.7</td>
</tr>
<tr>
<td>gBRDG</td>
<td>10</td>
<td>6</td>
</tr>
</tbody>
</table>

1 NUMBER 2 FALSE POSITIVE 3 FALSE NEGATIVE

4.4.3 Performance on Real-world Application

Our method is evaluated on the application of examining Electroencephalography (EEG) correlates of genetic predisposition to alcoholism [Frank and Asuncion (2010a)]. EEG records the brain’s spontaneous electrical activity by measuring the voltage fluctuations over multiple electrodes placed on the scalp. This technology has been widely used in clinical diagnosis, such as coma, brain death and genetic predisposition to alcoholism. In fact, encoded in the EEG data is a certain group structure, since each electrode records the electrical activity of a certain region of the scalp. Identifying and utilizing such spatial information has the potential of increasing stability of a prediction.

The training set contains 200 samples of 16384 dimensions, sampled from 64 elec-
trodes placed on subject’s scalp at 256 Hz (3.9-msec epoch) for 1 second. Therefore, the data can naturally be divided into 64 groups of size 256. We apply the lasso, group lasso, sparse group lasso, group SCAD, group MCP, group bridge, composite MCP and our proposed method on the training set and adapt the 5-fold cross-validation for selecting tuning parameters. More specifically, for lasso and group lasso, the candidate tuning parameters are specified by 10 parameters sampled using the logarithmic scale from the parameter spaces, while for the sparse group lasso, the parameters form a $10 \times 10$ grid, sampled from the parameter space in logarithmic scale. For our methods, the number of groups is selected from the set: $s_2 = \{30, 40, 50\}$ and $s_1$, the number of features is chosen from the set $\{50s_2, 100s_2, 150s_2\}$. The R package `grpreg` (80 parameters, 10-fold cross validation) are applied to other non-convex methods. The accuracy, sensitivity and specificity of classification together with the number of selected features and groups over a test set, which also contains 200 samples, are reported in Table 4.4. Clearly our methods achieve the best classification performance. Note that, although lasso’s performance is comparable with ours with even less features, however, it fails to identify the underlying group structure in the data, as revealed by the fact that all 64 groups are selected. Moreover, other nonconvex approaches such as the group SCAD, group MCP and group bridge seem to over-penalize the group penalty, which results in very few selected groups and suboptimal performance.

4.5 Summary

This chapter discusses a novel interpretable sparse group feature selection method, which is motivated from the ideal formulation of discrete feature and group selection.

$^3\lambda_{lasso} = \logspace(10^{-3}, 1)$, $\lambda_{glasso} = \logspace(10^{-2}, 1)$

$^4$The product space of $\lambda_{lasso} \times \lambda_{glasso}$
Table 4.4: Comparison of Performance on EEG Data. DC Which Stand for Difference of Convex functions, Denotes Our Proposed Method.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Acc¹</th>
<th>Sen²</th>
<th>Spe³</th>
<th># Feature</th>
<th># Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>lasso</td>
<td>67.0</td>
<td>72.0</td>
<td>62.0</td>
<td>2060</td>
<td>64</td>
</tr>
<tr>
<td>glasso</td>
<td>62.5</td>
<td>66.0</td>
<td>59.0</td>
<td>8704</td>
<td>34</td>
</tr>
<tr>
<td>sglasso</td>
<td>65.5</td>
<td>68.0</td>
<td>63.0</td>
<td>4834</td>
<td>61</td>
</tr>
<tr>
<td>DC</td>
<td>68.0</td>
<td>68.0</td>
<td>68.0</td>
<td>3890</td>
<td>25</td>
</tr>
<tr>
<td>gSCAD</td>
<td>60.5</td>
<td>59.0</td>
<td>62.0</td>
<td>1792</td>
<td>7</td>
</tr>
<tr>
<td>gMCP</td>
<td>60.5</td>
<td>59.0</td>
<td>62.0</td>
<td>256</td>
<td>1</td>
</tr>
<tr>
<td>cMCP</td>
<td>65.5</td>
<td>68.0</td>
<td>60.0</td>
<td>57</td>
<td>33</td>
</tr>
<tr>
<td>gBrdg</td>
<td>51.5</td>
<td>51.0</td>
<td>52.0</td>
<td>80</td>
<td>2</td>
</tr>
</tbody>
</table>

¹ accuracy  ² sensitivity  ³ specificity

Unlike traditional regularization based feature learning method, the model selection and parameter tuning procedure are greatly simplified as prior information can be effectively incorporated into the modeling through constraints. An efficient optimization scheme is developed based on the DC programming, accelerated gradient method and efficient projection. The efficiency and efficacy of the proposed method are validated on both synthetic data and real-world applications. The current method approximates the discrete constraints with continuous computational surrogate, which introduces extra computational cost. In the next chapter, we will focus on the discrete constraints directly and try to develop more efficient algorithms.
Chapter 5

INTERPRETABLE BI-LEVEL SELECTION: DISCRETE APPROACHES

5.1 Introduction

In this chapter, we further investigate interpretable bi-level selection methods based on the primitive model \[4.1\]. As mentioned in the previous chapter, the discrete nature of Model \[4.1\] prevents us from solving the combinatorial problem exactly in reasonable time. To overcome such a challenge, current research mainly falls into two categories. The first one focuses on finding suitable continuous computational surrogates for the discrete functions. This leads to various convex and nonconvex optimization models and our work in the previous chapter follows exactly the same spirit. On the other hand, instead of finding suitable continuous surrogates, computing a local solution of the discrete optimization problem directly also receives plenty of attention. The iterative hard thresholding (IHT) [Blumensath and Davies (2008, 2009)], orthogonal matching pursuit [Tropp and Gilbert (2007)] and group orthogonal matching pursuit [Lozano et al. (2009)] belong to this category. Although the optimization is by nature nonconvex, the efficiency of these algorithms is usually comparable (if not better) to that of convex relaxation models. However, to the best of our knowledge, these algorithms are proposed for feature selection only or group selection only. Whether they can be extended to handle bi-level selection properly and efficiently has not been much explored. In this chapter, we fulfill such a gap by introducing a hard thresholding model that is capable of bi-level selection. Our main contributions are: (1) we propose a novel bi-level selection model and show that the key combinatorial problem admits a globally optimal solution using dynamic programming; (2) we pro-
vide an error bound between our solution and the globally optimal one under the RIP (Restricted Isometry Property) theoretical framework Candes and Tao (2005); Cand
de (2008). We have evaluated the proposed algorithm on synthetic and real data. Results show that the proposed algorithm demonstrates encouraging performance while keeping comparable computational cost to convex relaxation models.

The remaining of the chapter is organized as follows: We present our algorithm for Problem (4.1) and discuss different variants in Section 5.2. In Section 5.3, we investigate a key sub-problem in our method and propose a dynamic programming algorithm that finds an optimal solution. The convergence property of the overall optimization framework is discussed in Section 5.4 and we present extensive empirical evaluation in Section 5.5. Section 5.7 summarizes the chapter. For notations, we mainly follow the symbols introduced in Eq. (4.1), i.e., $A$ stands for the design (sample) matrix, $y$ is the response, $x_{G_i}$ represents the regression model restricted on the $i$th group and $f$ denotes the objective function.

5.2 Optimization Algorithms

Motivated by the iterative hard thresholding algorithm for $\ell_0$-regularized problems Blumensath and Davies (2008), we adopt the Iterative Shrinkage and Thresholding Algorithm (ISTA) framework and propose the following algorithm for solving Problem (4.1):

In the proposed algorithm above, $f$ denotes the objective function and the “SGHT” in Algorithm 7 stands for the following Sparse Group Hard Thresholding (SGHT)
Algorithm 6 ISTA with Sparse Group Hard Thresholding

Input: A, y, s_1, s_2, \eta > 1

Output: solution x to Problem (4.1)

1: Initialize x^0.
2: for m \leftarrow 1, 2, \cdots do
3: Initialize L
4: repeat
5: \quad x^m \leftarrow \text{SGHT}(x^{m-1} - \frac{1}{L} \nabla f(x^{m-1}))
6: \quad L \leftarrow \eta L
7: until line search criterion is satisfied
8: if the objective stops decreasing then
9: \quad return x^m
10: end if
11: end for

problem with v as the input:

\begin{align*}
\text{minimize}_{x} \quad & \frac{1}{2} \| x - v \|_2^2 \\
\text{subject to} \quad & \sum_{j=1}^{p} I(|x_j| \neq 0) \leq s_1 \\
& \sum_{j=1}^{|G|} I(\| x_{G_j} \|_2 \neq 0) \leq s_2.
\end{align*}

(5.1)

Like most ISTA-based optimization algorithms, it is of critical importance that we can compute the projection step accurately and efficiently. In our case, the key part is exactly the SGHT problem. Although there are well established results on hard thresholding algorithms for \ell_0-regularization, adding one more constraint on group cardinality greatly complicates the problem and requires deeper analysis. We will present detailed discussion on how to compute an optimal solution to this problem
efficiently in the next section. Before that, we first introduce several possible variants of the proposed method. Notice that the target of Algorithm 6 is a nonconvex optimization problem. Different strategies for initialization and step-size may not only provide different convergence behavior, but also lead to a completely different solution. We consider three aspects in this work: step-size initialization, line search criterion and acceleration option.

5.2.1 Step-size Initialization

To provide an initial value of the step-size (Line 6. in Algorithm 6), we consider two strategies: a constant value and the Barzilai-Borwein (BB) method Barzilai and Borwein (1988). The BB method essentially finds the best multiple of identity matrix to approximate the Hessian matrix such that the least squares error of the secant equation is minimized, i.e., $L^k$ is initialized to

$$
\alpha^k = \arg \min_{\alpha} \| \alpha(x^k - x^{k-1}) - (\nabla f(x^k) - \nabla f(x^{k-1})) \| \\
= \frac{(\Delta g)^T(\Delta x)}{\|\Delta x\|^2}
$$

with a safeguard bound, where $\Delta g = \nabla f(x^k) - \nabla f(x^{k-1})$ and $\Delta x = x^k - x^{k-1}$. In this work, we set $L^k = \max(1, \alpha^k)$.

5.2.2 Line Search Criterion

We consider two line search termination criteria in this work, which we name as Lipschitz criterion and sufficient decrease criterion. Specifically the Lipschitz criterion finds the smallest $L$ that the following inequality is satisfied:

$$
f(x^k) \leq f(x^{k-1}) + \langle \nabla f(x^{k-1}), x^k - x^{k-1} \rangle + \frac{L}{2} \|x^k - x^{k-1}\|^2.
$$

On the other hand, the sufficient decrease criterion aims to find the smallest $L$
such that:

$$f(x^k) \leq f(x^{k-1}) - \frac{L\delta}{2} \|x^k - x^{k-1}\|^2_2.$$  \hspace{1cm} (5.3)

Inequality (5.2) is the standard way for $\ell_1$-regularized optimization \cite{beck2009fast} and is applied extensively in structured sparse learning \cite{liu2009regularized}. Inequality (5.3) and its variants are favored by most of the recent investigations on nonconvex regularized problems \cite{birgin2000algorithm, nesterov2007lagrange, beck2009fast}.

### 5.2.3 Acceleration Option

The ISTA framework has been shown to possess a convergence rate of $O(1/k)$ for a class of $\ell_1$-regularized/constrained optimization problems and can be further improved to $O(1/k^2)$ via adding a carefully designed search point \cite{nestro,beck2009fast}. However, whether the same strategy still works or makes the optimization diverge in the regime of nonconvex optimization remains unknown. In this work we consider both of them and retain the notation of FISTA \cite{beck2009fast} to denote the ISTA with the acceleration trick. See Algorithm \ref{alg:ista} for more detail about our FISTA.

**Table 5.1:** Specific Settings for Each Variant Considered in the Work. The Last Two Columns Denote the Lipschitz and Sufficient Decrease Line Search Criterion Respectively.

<table>
<thead>
<tr>
<th>Variants</th>
<th>FISTA</th>
<th>ISTA</th>
<th>BB</th>
<th>CONST</th>
<th>LIPS</th>
<th>DEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISTA</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ISTA-L</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>FISTA</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>FISTA-C</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.1 summaries different variants we consider in this work. All these variants will be examined in our experiments. We conclude this section by presenting several
Algorithm 7 FISTA with Sparse Group Hard Thresholding

Input: \( A, y, s_1, s_2, \eta \geq 1 \)

Output: solution \( x \) to Problem (4.1)

1: Initialize \( x^{-1}, x^0, \alpha^{-1} \leftarrow 0, \alpha^0 \leftarrow 1 \)

2: for \( m \leftarrow 1, 2, \cdots \) do

3: \( \beta^m \leftarrow \frac{\alpha^m - 1}{\alpha^{m-1}} \)

4: \( u^m \leftarrow x^{m-1} + \beta^m (x^{m-1} - x^{m-2}) \)

5: Initialize \( L \)

6: repeat

7: \( x^m \leftarrow \text{SGHT}(u^m - \frac{1}{L} \nabla f(u^m)) \)

8: \( L \leftarrow \eta L \)

9: until line search criterion is satisfied

10: if the objective stops decreasing then

11: return \( x^m \)

12: end if

13: end for

additional features of the proposed algorithm.

Remark 1. One significant advantage of adhering to the discrete model is that incorporating prior knowledge about the grouping structure is quite straightforward. Remember that the two parameters in our model are just the upperbound of features and feature groups respectively. In addition, model selection procedures such as cross-validation can be greatly facilitated since we only need to consider integer values, which are often quite small in real-world applications. On the contrary, the regularizers in most of the existing works are real-valued and may not provide much insights for parameter-tuning.

Remark 2. Although we consider our bi-level learning model in a linear regression
setting, the technique can be readily extended to more general problems by choosing appropriate loss functions. Particularly, in order to extend our model to classification tasks, the widely-used logistic loss function can be applied instead of the least squares function in Eq. (4.1) and the proposed Algorithm 6 can be applied by changing the procedure that computes the gradient. In general, the proposed model can be extended to any convex loss functions with a simple gradient computation.

5.3 Optimal Solution of SGHT

In this section, we show how to solve the SGHT problem in Eq. (5.1) efficiently using dynamic programming. Before presenting our algorithm, we first explore some key properties of Problem (5.1). As highlighted previously, the major challenge comes from the two coupled constraints. Therefore, we first consider the special case where only one of the two constraints is present. Some straight-forward analysis leads to the following results:

**Lemma 3.** If only the cardinality constraint is present, the optimal solution of Problem (5.1) can be obtained by setting the \( p - s_1 \) smallest (in absolute value) elements of \( \mathbf{v} \) to zero. Similarly for group cardinality constraint, it suffices to find the \( |G| - s_2 \) smallest groups (in \( \ell_2 \)-norm) and set them to zero.

Based on Lemma 3, it is also easy to verify that for any optimal solution \( \mathbf{x}^* \) of Problem (5.1), each element \( x_i^* \) is either equal to \( v_i \) or zero, where the subscript \( i \) denotes the \( i \)th element of the vector. Therefore we have the following proposition providing an equivalent but discrete characterization of the original SGHT problem:

**Proposition 3.** Finding the optimal solution of problem (5.1) is equivalent to the following **Sparse Group Subset Selection (SGSS) problem:**

Given a set \( S \) on which a nonnegative value function \( f \) is defined. \( C = \{C_1, C_2, \cdots , C_{|G|}\} \)
is a collection of disjoint subsets of $S$ such that $S = \bigcup_{i=1}^{|G|} C_i$. Find a subset $S' \subset S$ with the maximum value such that the cardinality of $S$ is no more than $s_1$ and $S'$ has nonempty intersections with at most $s_2$ elements from $C$. The value of a subset is defined as the summation of all the values of its elements.

We claim that the SGHT has an optimal solution if and only if we can find an optimal solution for the SGSS problem. We provide a one-way reduction (the “if” part) here. The other way is almost identical. The original SGHT problem can be reduced to SGSS by simply setting $S = \{1, 2, \cdots, p\}$ with the value function defined as $f(i) = v_i^2$ for all $1 \leq i \leq p$ and $C_i = G_i$ for all $1 \leq i \leq |G|$. Suppose $S'$ is the optimal solution of SGSS. Then the optimal solution of SGHT can be readily obtained via:

$$x^* = \begin{cases} 
    v_i & \text{if } i \in S' \\
    0 & \text{otherwise.}
\end{cases} \quad (5.4)$$

In the sequel, we will focus on the SGSS problem and provide an efficient algorithm to compute its globally optimal solution. The term cardinality and group cardinality are used to characterize the size of $S'$ and the number of elements from $C$ with which $S'$ has a nonempty intersection, respectively.

Let $T(i, j, k)$ denote the maximum value we can obtain by choosing a subset $S'$, whose cardinality is no more than $k$ and group cardinality is at most $j$. In addition, $S'$ is only allowed to have nonempty intersection with $C_1, C_2, \cdots, C_i$. Therefore $T$ is in essence a three-dimensional table of size $(|G| + 1) \times (s_2 + 1) \times (s_1 + 1)$ (the table is zero-indexed). It is easy to verify that, if we are able to compute all the values in table $T$ correctly, the maximum value one of the SGSS problem is given by $T(|G|, s_2, s_1)$.

Next we propose a dynamic programming algorithm to compute the table $T$. The motivation behind our method is the existence of optimal substructure and overlapping subproblems [Leiserson et al. (2001)], two major ingredients for an efficient dy-
namic programming algorithm. More specifically, when we try to compute \(T(i, j, k)\), the optimal solution must fall into one of the two situations: whether the \(C_i\) is selected or not. If not, we can simply conclude that \(T(i, j, k) = T(i - 1, j, k)\). On the other hand, if \(C_i\) is selected, we need to determine how many elements from \(C_i\) are included in the optimal solution. Suppose the optimal solution takes \(t\) elements from \(C_i\), then we must have \(T(i, j, k) = T(i - 1, j - 1, k - t) + CH(i, t)\), where \(CH(i, t)\) denotes the maximum value one can get from choosing \(t\) elements out of \(C_i\). The optimal \(t\) can be computed via enumeration. To sum up, the computation of \(T(i, j, k)\) can be written in the following recursive form:

\[
T(i, j, k) = \max \left\{ T(i - 1, j, k), \max_{1 \leq t \leq \min(k, |G_i|)} T(i - 1, j - 1, k - t) + CH(i, t) \right\}
\]

It is clear from above that \(T(i, j, k)\) can be computed using only the values in the table \(T\) with smaller indices. Therefore we can compute each element of the table \(T\) in increasing order for each index; see Figure 5.1 for more detail. In addition, to further reduce the complexity, function \(CH(i, t)\) can be precomputed before the dynamic programming process. We present the detailed description of the proposed method in Algorithm 8. From table \(T\), we are able to calculate the minimum objective value of the SGHT problem, which is exactly \(\frac{1}{2}(\|v\|_2^2 - T(|G|, s_2, s_1))\). In order to calculate the optimal solution \(x^*\), all we need to know is the indices of selected elements in \(S\) and the optimal solution can be constructed through Eq. (5.4). We compute such information by adding one table \(P\) (stands for path) in the proposed algorithm. Specifically, \(P(i, j, k) = 0\) means the \(C_i\) is not selected in the computation of \(T(i, j, k)\). Otherwise we set

\[
P(i, j, k) = \arg \max_{1 \leq t \leq \min(k, |G_i|)} T(i - 1, j - 1, k - t) + CH(i, t),
\]

which is just the number of selected features in the \(i\)th group \((C_i)\) in the optimal
solution. To recover the indices of all the selected elements, we will start from \( P(|G|, s_2, s_1) \) with a backtracking procedure and record the number of selected elements in each group. Algorithm 6 provides a formal description of this process. It accepts the table \( P \) as input and returns the \( cnt \) table which contains the number of selected elements in each group. Finally computing the optimal \( \mathbf{x}^* \) only amounts to keeping the top selected elements for each group and setting the remains to zero.

Figure 5.1: Illustration of the Order of Computation for Each Element in \( T \). While Computing \( T(i, j, k) \), We Only Need Values in Those Red Squares, Which Are Located in the Previous Rectangle (in Terms of \( i \)-Axis) and of Equal or Smaller Coordinates on Axes \( j \) and \( k \). Therefore the Computation Can be Naturally Carried Out in Three Nested Loops, One for Each Axis Respectively.

We analyze the time complexity of our proposed algorithm as follows. Notice that the time needed to precompute the table \( CH \) is given by:

\[
O\left( \sum_{i=1}^{|G|} |G_i| \log(|G_i|) \right) = O(p \log p),
\]
the dynamic programming part for computing both $T$ and $P$ takes

$$O\left(\sum_{i=1}^{|G|} s_2 s_1 |G_i|\right) = O(s_1 s_2 \sum_{i=1}^{|G|} |G_i|) = O(ps_1 s_2),$$

and the backtracking needs clearly $O(|G|)$ operations. Therefore the overall time complexity is

$$O(p(s_1 s_2 + \log p) + |G|) = O(s_1 s_2 p + p \log p).$$

When the number of features and feature groups selected is small, the SGHT problem can be solved efficiently.

**Remark 3.** After the publication of our work [Xiang et al. (2014)], we are aware of Baldassarre et al.’s working paper [Baldassarre et al. (2013)], in which they consider a special overlapping group structure. These two works are done independently and both of them can be applied to solve the proposed SGHT problem.

### 5.4 Convergence Analysis

In this section, knowing that the key SGHT sub-problem can be efficiently computed, we assess the quality of the solution produced by the overall optimization procedure (Algorithm 1). Specifically, since the constraints of Eq. (4.1) are nonconvex and only a local minimum can be found through our proposed method, we are interested in studying how close (in terms of Euclidean distance) the obtained solution to the optimal solution of the optimization problem (4.1). Although we are not aware of the optimal solution, the bound between our solution and the optimal one can be analyzed under the theoretical framework of restricted isometry property (RIP) [Candès and Tao (2005)]. A matrix $A \in \mathbb{R}^{n \times p}$ is said to satisfy the RIP property with constant $\delta_s$ if the following property holds for any $s$-sparse vector $x$, i.e.,
\[ \|x\|_0 \leq s: \]
\[ (1 - \delta_s)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta_s)\|x\|_2^2. \]

The RIP constant essentially assesses the extent to which the given matrix resembles an orthogonal matrix and theoretical analyses often require certain upperbound on the RIP constant. It is easy to see that \( \delta_s \) is non-decreasing w.r.t \( s \) and a smaller value of \( \delta_s \) indicates more rigid conditions we require from \( A \). In order to apply the RIP based analysis for our method, a group-RIP constant is introduced to incorporate the group structure. Matrix \( A \) has a group-RIP constant \( g \) if for any vector \( x \) that spans no more than \( g \) groups, i.e., \( \sum_{j=1}^{G} I(\|x_{G_j}\|_2 \neq 0) \leq g \), the following relation are satisfied:

\[ (1 - \delta_g)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta_g)\|x\|_2^2. \]

Our next result provides an error bound between an optimal solution of Problem (4.1) and the solution given by our proposed Algorithm 6 with \( L \) fixed to 1.

**Theorem 5.** Let \( x^* \) be a globally optimal solution of Problem (4.1) and \( x^k \) be the solution we obtain after the \( k \)th iteration in Algorithm 6 with \( L = 1 \). If \( c_1 < \frac{1}{2} \), the following result holds:

\[ \|x^k - x^*\|_2 \leq (2c_1)^k\|x^0 - x^*\|_2 + \frac{2\sqrt{1 + c_2}}{1 - 2c_1}\|e^*\|_2, \]

where \( e^* = y - Ax^* \), \( c_1 = \min\{\delta_{3s_1}, \delta^{3s_2}\} \), \( c_2 = \min\{\delta_{2s_1}, \delta^{2s_2}\} \). In addition, if \( c_2 < \frac{1}{4} \), it is also true that:

\[ \|x^k - x^*\|_2 \leq (4c_2)^k\|x^0 - x^*\|_2 + \frac{2\sqrt{1 + c_2}}{1 - 4c_2}\|e^*\|_2. \]

Theorem 6 clearly shows that the parameter estimation error of the proposed algorithm decreases linearly (with coefficient of \( 2c_1 \) or \( 4c_2 \)) till a fixed error term is met. In addition, such an error term is proportional to the prediction error of the optimal
solution of Problem (1.1). The proof of Theorem 5 mainly utilizes the technique in Foucart (2012) and the details are left in the Appendix. We provide an illustrative example of the convergence procedure in Figure 5.2: if the assumptions on the (group) RIP constant hold, the sequence generated by running our algorithm is guaranteed to converge into a region centered at $x^*$ with radius at most $c\|e^*\|_2$, where $c$ is a constant. As we can observe from Figure 5.2 and Theorem 5, the difference between the unknown globally optimal solution of Problem (1.1) and ours is upperbounded by a multiple of the underlying error term $\|e^*\|_2$. In addition, such a difference cannot be canceled unless we have $e^* = 0$, in which case Theorem 5 essentially states that our method admits a linear convergence rate Nocedal and Wright (2000).

![Figure 5.2: Illustration of the Convergence Behavior of the Proposed Algorithm. The Parameter Estimation Error Decreases Linearly Before Entering Into a Region Centered at $x^*$ With Radius Proportional to the Prediction Error of $x^*$.](image)

5.5 Experiments

5.5.1 Evaluation of SGHT

Recall that solving SGHT (Problem (1.1)) accurately and efficiently is the key to our optimization procedure (Algorithm 6). We have theoretically analyzed the
correctness and time complexity of our method in Section 5.3. In this part, we present empirical studies on the efficiency of our proposed Algorithm 8. As we have analyzed previously, three factors including the number of candidate features, the number of selected groups and the number of selected features determine the time complexity. We conduct the evaluation in four different scenarios, each of which demonstrates the relationship between the running time and some particular factors while keeping other factors unchanged. Specific settings are listed in Table 5.2.

Table 5.2: Experiment Setup for Evaluation of SGHT

<table>
<thead>
<tr>
<th>Fixed variable</th>
<th># Group</th>
<th># Feature</th>
<th>$s_1$</th>
<th>$s_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario 1</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scenario 2</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scenario 3</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Scenario 4</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Scenario 1. Varying number of features $p$ with incremental candidate set.** We vary the number of features $p$ from 1,000 to 5,000,000. The number of groups is fixed to 100 in this case, i.e., $|G| = 100$. $s_2$ is set to 20%, 40% and 60% of the total number of groups respectively and the value of $s_1$ is set to $5s_2$, i.e., we want to approximately select 5 features per group.

- **Scenario 2. Varying number of groups $|G|$ with incremental candidate set.** $p$ is fixed to 1,000,000 and $G$ is chosen from the set of $\{10, 50, 100, 150, 200\}$. The value of $s_1$ and $s_2$ is set according to the same strategy in Scenario 1.

- **Scenario 3. Varying number of groups $|G|$ with fixed candidate set.** We conduct this evaluation in order to verify our theoretical result that the number of groups $|G|$ is not a dominating factor of time complexity. Specifically we fix
the value of \( p \) to 1,000,000 and choose \(|G|\) from \{50, 100, 500, 1000, 5000, 10000\}. \( s_1 \) and \( s_2 \) are fixed as 50 and 5 respectively.

- Scenario 4. Incremental candidate set with fixed number of groups and features. In this case, 1,000,000 variables are partitioned into 100 groups of equal size. We attempt to select 10% \( \sim \) 60% of all the groups and approximately 20 features per group.

Figure 5.3 demonstrates the running time (in seconds) of our SGHT algorithm of all four scenarios. Specifically, the nearly flat curve in our third experiment corroborates with the theoretical result that the number of groups is not a major factor of the time complexity. In other cases, our algorithm exhibits its capability of handling large-scale applications. Particularly, when only a small number of features and feature groups are wanted, as is the common situation in high-dimensional variable selection, our algorithm is capable of computing a globally optimal solution for SGHT with a performance competitive to its convex computational surrogate such as the soft-thresholding Donoho (2002).

### 5.5.2 Evaluation of Convergence

We study the convergence behavior of different implementations of our discrete optimization approach proposed in Section 5.4. The evaluation is carried out on a collection of randomly generated data sets \((A, y)\). Specifically, we generate \( A \in \mathbb{R}^{n \times p}, \ y \in \mathbb{R}^n \), where the values of \( n \) and \( p \) are chosen from the following set:

\[
\{(100, 2000), (100, 5000), (1000, 20000), (1000, 50000)\}.
\]

All of the \( p \) features are partitioned into groups of size 100. The value of \( s_2 \) is selected from \{0.1|G|, 0.2|G|\}, i.e., we select 10% and 20% groups. \( s_1 \) is set to \( 5s_2 \), which leads to the effect of within-group sparsity.
For all of the variants, we terminate the programs when either the relative change of objective value in two consecutive iterations or the gradient of the objective is less than a given threshold. The objective values of up to the first 100 iterations as well as the running time for each variant are reported in Figure 5.4. The results demonstrate the effect of BB to initialize the step-size. Both ISTA with lipschiz line search criterion (blue in Figure 5.4) and FISTA (black in Figure 5.4) deliver superior performance, particularly for large data sets and large number of selected groups/features.

### 5.5.3 Simulation Results

We examine the proposed bi-level method on synthetic data which consist of both group selection and bi-level variable selection. The data generation follows the
Figure 5.4: Convergence Results of Different Variants of the Proposed Discrete Optimization Approach on Synthetic Data, Where ISTA-L and FISTA-C Stand for ISTA with Lipschitz Line Search Criterion and FISTA with Const Step-size Initialization. All The Algorithms Are Evaluated on Four Data Sets, from Top to Bottom, of Which the Size of $A$ Is (100,2000), (100,5000), (1000,20000) and (1000,50000) Respectively. The Number of Selected Group ($s_2^2$) is Chosen from 0.1$|G|$ and 0.2$|G|$ and The Corresponding Results Are Listed from Left to Right. For Each Parameter Setting, We Report the Objective Values up to 100 Iterations (The Lines) As Well As the Running Time in Seconds (The Histograms).

procedures recommended in the literature Yuan and Lin (2006); Xiang et al. (2013c): the data set is generated via the linear model $y = Ax + \epsilon$, where both of the design matrix $A \in \mathbb{R}^{100 \times 200}$ and the noise term $\epsilon$ follow a normal distribution. The ground truth $\bar{x}$ is partitioned into 20 groups of equal size. In addition, two kinds of grouping structure are considered in this experiment; see Figure 5.5 for more detail. The goal is to obtain an accurate (in terms of least squares) estimator of $\bar{x}$ that also preserves the grouping structure, given only $A$ and $y$.

State-of-the-art bi-level feature learning algorithms, including the convex sparse group lasso, two fractional models Xiang et al. (2013d) (frac(1,2) for bi-level variable selection and frac(2,1) for group selection) and DC approximation approach Xiang
Figure 5.5: Illustration of the Grouping Effect in the Ground Truth Model \( \tilde{\mathbf{x}} \). Both Cases Include Redundant Groups (Group 7 to Group 20). In Addition, The First Case Contains a Bi-level Sparsity. The Values Within Each Group Are Identical, As Shown in the Color Map.

\textit{et al.} (2013a), are included for comparison. It is worth mentioning that the DC approach deals with exactly the same formulation as ours but resort to using continuous computational surrogate. In addition, we also include orthogonal matching pursuit (OMP) and group orthogonal matching pursuit (gOMP) in the experiments as they provide baseline results for discrete optimization approach. For both fractional models, we choose 5 regularizers from the interval \([10^{-8}, 10^2]\). For DC approach and our method, \( s_2 \) is selected from \( \{2, 4, 6, 8, 10\} \) and \( s_1 \) is chosen from the set of \( \{2s_2, 4s_2, 6s_2, 8s_2, 10s_2\} \). Since the parameters of OMP and gOMP are just the number of selected features and feature groups respectively, we set \( \{6, 12, 18, \cdots, 60\} \) as the candidate parameter set for OMP and similarly \( \{2, 4, 6, \cdots, 10\} \) for gOMP. Five-fold cross-validation is carried out to choose the best parameter for each method. The tuned models are then tested on an i.i.d testing set. Following the setups in previous work \textit{Breheeny and Huang} (2009); \textit{Xiang et al.} (2013a,b), the number of selected groups/features, the number of false positive selections and false negative selections and the running time (in seconds) are reported in Table 5.3. We can observe that the approaches with discrete parameters (OMP, gOMP, DC approach and our method) deliver more accurate estimation on the number of groups and features, compared to regularization-based approaches. Particularly, our method demonstrates the best performance in the bi-level selection tasks and is second only to gOMP in the sce-
nario of group selection. The low false positive rate means that redundant groups are effectively screened. However, this could lead to a relatively high but still reasonable false negative rate. Such a phenomenon is also observed in existing work [Breheny and Huang (2009)]. As of efficiency, it is expected that OMP and gOMP are the most efficient methods due to their cheap and small number of iterations. Among others, our method requires the least amount of running-time. In addition, the DC approach, which needs to refine the continous surrogate within each iteration, requires the most computational effort (nearly twice of the time of our method).

5.5.4 Real-world Applications

We conclude the experiment section with a study on the Boston Housing data set [Frank and Asuncion (2010b)]. The original data set is used as a regression task which contains 506 samples with 13 features. Furthermore, to take into account the non-linear relationship between variables and response, up to third-degree polynomial expansion is applied on each feature, as suggested in previous works [Swirszcz et al. (2009)]. Specifically, for each variable $x$, we record $x$, $x^2$ and $x^3$ in the transformed data and gather them into one group. We randomly take 50% of the data as the training set and leave the rest for testing. The parameter settings for each method follow the same spirit in our last experiment and are properly scaled to fit this data set. We fit a linear regression model on the training data and report the number of selected features, feature groups as well as the mean squared error (MSE) on the testing set in Table 5.4. Five-fold cross validation is adopted for parameter tuning and all the results are averaged over 10 replications. We can observe from the table that our method shows the best prediction results with the least amount of features and feature groups.
5.6 Extension to Fused Hard Thresholding Models

Motivated by the investigation on hard thresholding as well as our research on discrete model for sparse group feature selection, we try to extend the hard thresholding models to more complicated but useful case, following the pathway on which sparse learning research was carried out. Particularly, we consider the fused lasso penalty here, which is defined as \( R(x) = \sum_{i=2}^{p} |x_i - x_{i-1}| \), where \( x \in \mathbb{R}^p \). Fused lasso penalty can be beneficial when a smooth change of elements is expected\(^{(2005)}\). However, this convex regularization term penalizes not only the number of elements change but also the magnitude. Also, it inherits one limitation of sparsity-inducing penalties: unclear quantitative relation between regularization and number of selected features. We propose to apply the following cardinality constraint \( \sum_{j=2}^{p} I(x_j \neq x_{j-1}) \leq s_f \) to handle these issues.

As in the sparse group hard thresholding, we can readily employ the ISTA framework as long as the following fused hard thresholding problem can be solved accurately:

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \| x - v \|_2^2 \\
\text{subject to} & \quad \sum_{j=1}^{p} I(|x_j| \neq 0) \leq s_1 \\
& \quad \sum_{j=2}^{p} I(x_j \neq x_{j-1}) \leq s_f.
\end{align*}
\]

Similar to the sparse group hard thresholding problem, we consider the dynamic programming approach for this discrete optimization problem and transform the minimization formulation to a subset selection problem. Specifically, let \( T(i, j, k) \) denotes the minimal objective value one could achieve by selecting no more than \( k \) of the first \( i \) variables, such that no more than \( j \) value groups exist. The recursive form can be
established as follows:

\[
T(i, j, k) = \min \left\{ \min_{1 \leq t \leq \min(k, i)} T(i - t, j - 1, k - t) + SE(i, t), \min_{1 \leq t \leq i} T(i - t, j - 1, k) + NM(i, t) \right\}
\]

The formula considers forming the sub-vector \(x(i - t + 1 : i)\) (we adopt the matlab style notation) as a value group. The first case attempts to build a non-zero group and therefore the optimal solution would be assigning every element to be the mean value of \(x(i - t + 1 : i)\) and the cost is the corresponding squared error \((SE(i, t))\). On the other hand, if we consider \(x(i - t + 1 : i)\) as a zero value group, then the cost is just the \(\|x(i - t + 1 : i)\|_2^2\), i.e., \(NM(i, t)\) in the formula. It is straightforward to see that the time complexity is \(O(p^2s_1s_f)\). Since both \(SE\) and \(NM\) can be calculated on the fly, the space requirement is \(O(ps_1s_f)\).

5.7 Summary

Based on the work in the previous chapter, we continue to study interpretable models for simultaneous feature and feature group selection. Unlike previously developed methods which are based on continuous computational surrogate for the discrete selection problem, we focus on the discrete model directly. The main contribution is that we transform the key proximal part to the sparse group subset selection problem and present a dynamic programming algorithm which is capable of finding a global optimum. The projection is then fed into the Iterative Shrinkage and Thresholding Algorithm (ISTA) framework to produce a local solution for the original problem. Systematic investigations are carried out on optimization algorithms, convergence property as well as empirical evaluations. The proposed model delivers superior performance in both group selection and bi-level variable selection settings and possesses significant advantage on efficiency, particularly when only a small number of fea-
tures and feature groups are demanded. In addition, due to the discrete parameters, model selection procedures such as parameter tuning can be greatly facilitated. We also show how to extend this hard thresholding algorithm to handle the fused lasso penalty, in order to achieve a sparse and smooth model.
Algorithm 8 Dynamic programming algorithm for SGSS

Input: $S, C = \bigcup_{i=1}^{[G]} C_i, s_1, s_2$

Output: $T, P$

1: $T \leftarrow 0, CH \leftarrow 0, P \leftarrow 0$

2: for $i = 1$ to $|G|$ do

3: sort $C_i$ in decreasing order of magnitude

4: for $t = 1$ to $|G_i|$ do

5: $CH(i, t) \leftarrow CH(i, t - 1) + C_i(t)$

6: end for

7: end for

8: for $i = 1$ to $|G|$ do

9: for $j = 1$ to $s_2$ do

10: for $k = 1$ to $s_1$ do

11: $T(i, j, k) \leftarrow T(i - 1, j, k)$

12: for $t = 1$ to $G_i$ do

13: $w \leftarrow T(i - 1, j - 1, k - t) + CH(i, t)$

14: if $w > T(i, j, k)$ then

15: $T(i, j, k) = w$

16: $P(i, j, k) = t$

17: end if

18: end for

19: end for

20: end for

21: end for
Algorithm 9 Linear backtracking algorithm for finding the number of selected elements in each group

**Input:** $P$, $s_1$, $s_2$

**Output:** $cnt$

1. $j \leftarrow s_2$, $k \leftarrow s_1$

2. for $i = |G|$ downto 1 do

3. $cnt(i) \leftarrow P(i, j, k)$

4. if $cnt(i) > 0$ then

5. $j \leftarrow j - 1$

6. $k \leftarrow k - cnt(i)$

7. end if

8. end for
Table 5.3: Comparison of Performance on Synthetic Data. NO, FP and FN Denote Number, False Positive Number and False Negative Number Respectively. All the Results Are Averaged Over 10 Replications.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Bi-level Selection (case 1)</th>
<th>Group Selection (case 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Groups</td>
<td>Features</td>
</tr>
<tr>
<td></td>
<td>NO.</td>
<td>FP</td>
</tr>
<tr>
<td>sgLasso</td>
<td>19.10</td>
<td>13.10</td>
</tr>
<tr>
<td>frac(1,2)</td>
<td>8.90</td>
<td>2.90</td>
</tr>
<tr>
<td>frac(2,1)</td>
<td>8.60</td>
<td>2.80</td>
</tr>
<tr>
<td>OMP</td>
<td>8.40</td>
<td>3.00</td>
</tr>
<tr>
<td>gOMP</td>
<td>3.80</td>
<td>0.00</td>
</tr>
<tr>
<td>DC</td>
<td>7.70</td>
<td>2.00</td>
</tr>
<tr>
<td>sgHT</td>
<td>5.20</td>
<td>0.00</td>
</tr>
</tbody>
</table>
Table 5.4: Comparison of Performance on the Boston Housing Data Set. All the Results Are Averaged Over 10 Replications.

<table>
<thead>
<tr>
<th>Methods</th>
<th># Group</th>
<th># Feature</th>
<th>mse</th>
</tr>
</thead>
<tbody>
<tr>
<td>sgLasso</td>
<td>7.10</td>
<td>20.30</td>
<td>2603.50</td>
</tr>
<tr>
<td>frac(1, 2)</td>
<td>9.30</td>
<td>16.10</td>
<td>8485.12</td>
</tr>
<tr>
<td>frac(2, 1)</td>
<td>9.60</td>
<td>28.80</td>
<td>8530.00</td>
</tr>
<tr>
<td>OMP</td>
<td>4.30</td>
<td>6.00</td>
<td>8089.91</td>
</tr>
<tr>
<td>gOMP</td>
<td>4.20</td>
<td>12.00</td>
<td>8924.55</td>
</tr>
<tr>
<td>DC</td>
<td>2.70</td>
<td>5.20</td>
<td>8322.14</td>
</tr>
<tr>
<td>SGHT</td>
<td>2.10</td>
<td>3.00</td>
<td>545.27</td>
</tr>
</tbody>
</table>
In this chapter, I summarize my thesis work and highlight the contributions. In addition, I list some interesting directions for future research.

6.1 Summary of Contributions

With the advances of data-collecting technologies, learning from multiple heterogeneous data sources becomes increasingly popular in many areas such as bioinformatics, disease diagnosis and web mining. Motivated by the challenges arisen from these applications, such as multi-modality, high-dimensionality and existence of block-wise missing data, we conduct research on developing effective feature learning models. Particularly, special attention is paid to the following aspects: (1) information fusion from multiple heterogeneous data sources; (2) simultaneous feature and feature group selection (bi-level selection); (3) flexibility to handling block-wise missing data without imputation and (4) interpretable model selection.

We start with investigating bi-level learning on complete data. Inspired by the compressed sensing technique, we propose a unified bi-level selection model. The proposed model contains popular methods such as lasso, group lasso and $\ell_{1,\infty}$-regularization as special cases. Interestingly, some nonconvex models can also be derived and demonstrate superior performance compared to classical convex methods.

Block-wise missing data is frequently encountered in practical applications, but how to extend existing bi-level learning techniques to deal with block-wise missing data remains largely unexplored. In this thesis, we take the prediction of Alzheimer’s Disease as an example and propose a systematic study. Our contributions are two-
fold: (1) the proposed incomplete model avoids direct imputation of the missing data, and is capable of bi-level feature learning; (2) applying the proposed method to incomplete data requires solving nonconvex optimization problems. We present efficient optimization algorithms, to find the solution by solving a sequence of convex sub-problems. The proposed incomplete model learns a single model for each data source across different groups (each group corresponds to one data source combination), and learns the prediction model for each group by computing a weighted combination of the models (one model for each source) involved in the group, thus it provides out-of-sample prediction, overcoming the limitation of existing methods. We also evaluate the effectiveness of the proposed models, compared to existing methods using data from the Alzheimer’s Disease Neuroimaging Initiative (ADNI). A total of 780 subjects, who have at least one of the four major types of data (MRI, PET, CSF, and proteomics) available, were included in our study. Our experiments show the potential of the proposed models for analyzing multiple heterogeneous sources with block-wise missing data.

In order to achieve flexible control over the amount of selection, e.g., the number of selected variables and groups, we study two novel sparse group feature selection methods, based on continuous and discrete optimization respectively. Both of them are motivated from the ideal formulation of discrete feature and group selection. For the continuous approach, an efficient optimization scheme is developed based on the DC programming, accelerated gradient method and efficient projection. In addition, theoretical properties on the accuracy of selection and parameter estimation are analyzed. For the discrete approach, we transform the proximal part to the sparse group subset selection problem and present a dynamic programming algorithm which is capable of finding a global optimum. The projection is then fed into the Iterative Shrinkage and Thresholding Algorithm (ISTA) framework to produce a solution for
the original problem. The efficiency and efficacy of the two proposed methods are validated on both synthetic data and real-world applications.

6.2 Future Work

For further investigation, the following directions appear promising.

Large scale structured sparse learning based on sparsity-inducing penalties has received intensive investigations during the past decades. Besides popular methods such as lasso, group lasso, complicated models incorporating the structure information also attract great attention. Examples are fused lasso [Tibshirani et al. (2005)], tree-structured lasso [Liu and Ye (2010)] and overlapping group lasso [Yuan et al. (2011)]. In this thesis, we have shown that, as an extension of sparse group lasso, sparse group hard thresholding algorithm delivers comparable (if not better) performance quite efficiently, despite the internal optimization problem is nonconvex. It is interesting to further investigate this part and find out whether there exists hard thresholding based counterparts of complex structured sparse learning models. We have show such a possibility for fused lasso and it is expected that more complicated models can be extended under the hard thresholding framework.

In addition to the optimization algorithms, statistical properties are also of great interest. For classical compressed sensing models such as lasso, their statistical properties, e.g., the prediction error and parameter estimation error, are important quantitative metrics. How to derive these results for hard thresholding based algorithms and whether they are in theory comparable to those results of convex sparse learning models are worth studying.

Last but not least, I’m interested to see the proposed methods applied to more real-world applications involving the group structure.
REFERENCES


Duda, R., P. Hart and D. Stork, Pattern Classification (1997).


86


APPENDIX A

PROOF OF THEOREM 4
The proof uses a large deviation probability inequality of Wong and Shen (1993) to treat one-sided log-ratio likelihoods with constraints.

Let \( S = \{ x^r : \| x^r \|_0 \leq s_0^1, \| x^r \|_0, G \leq s_0^2 \}, \| x \|_0 = \sum_{j=1}^P I(|x_j| \neq 0) \) is the \( L_0 \)-norm of \( x \), and \( \| x \|_0, G = \sum_{j=1}^{|G|} I(\| x_j \|_2 \neq 0) \) is the \( L_0 \)-norm over the groups.

Now we partition \( S \). Note that for \( G \subseteq \{ G_1, \ldots, G_{|G|} \} \), it can be partitioned into \( G = (G \setminus G^0) \cup (G \cap G^0) \). Then

\[
S = \bigcup_{i=0}^{s_2^0} \bigcup_{G \in B_i} S_{BG},
\]

where \( S_{BG} = \{ x^r \in S : G(x) = G = (G_{i_1}, \ldots, G_{i_k}), \sum_j |B_{G_j}| \leq s_1^0 \}, \) and \( B_i = \{ G \neq G_0 : |G \setminus G| = i, |G| \leq s_0^2 \}, \) with \( |B_i| = \left( \frac{s_0^2}{s_2^0} \right) \sum_{j=0}^i \left( \frac{|G| - s_0^2}{j} \right) ; i = 0, \ldots, s_0^2. \)

To bound the error probability, let \( L(x) = -\frac{1}{2} \| Ax - y \|^2 \) be the likelihood. Note that

\[
\{ \hat{x} \neq \hat{x}^0 \} \subseteq \{ L(\hat{x}) - L(\hat{x}^0) \geq 0 \} \subseteq \{ L(\hat{x}) - L(x^0) \geq 0 \}.
\]

This together with \( \{ \hat{x} \neq \hat{x}^0 \} \subseteq \{ \hat{x} \in S \} \) implies that

\[
\{ \hat{x} \neq \hat{x}^0 \} \subseteq \{ L(\hat{x}) - L(x^0) \geq 0 \} \cap \{ \hat{x} \in S \}.
\]

Consequently,

\[
I \equiv P(\hat{x} \neq \hat{x}^0) \leq P\left( L(\hat{x}) - L(x^0) \geq 0; \hat{x} \in S \right)
\]

\[
\leq \sum_{i=1}^{s_2^0} \sum_{G \in B_i} \sum_{S_{BG}} P^*\left( \sup_{x \in S_{BG}} (L(x) - L(x^0)) \geq 0 \right)
\]

\[
\leq \sum_{i=1}^{s_2^0} \sum_{j=1}^{s_1^0} \sum_{|G| = i, |B_G| = j} P^*\left( \sup_{x \in S} (L(x) - L(x^0)) \geq 0 \right),
\]

where \( P^* \) is the outer measure and \( S = \{ -\log(1 - h^2(x, x^0)) \geq \max(i, 1)C_{\min}(x^0) - d_3 \tau d_2 p, x \in S_{BG} \} \). The last two inequalities use the fact that \( S_{BG} = \{ x \in S_{BG} : \max(|G^0 \setminus G|, 1)C_{\min}(x^0) \leq -\log(1 - h^2(x, x^0)) \} \subseteq S \), under Assumption (3).

For \( I \), we apply Theorem 1 of Wong and Shen (1993) to bound each term. Towards this end, we verify their entropy condition (3.1) for the local entropy over \( S_{BG} \) for \( |G| = 1, \ldots, s_2^0 \) and \( |B_G| = 1, \ldots, s_1^0 \). Under Assumption (2) \( \varepsilon = \varepsilon_{n,p} = (2c_0)^{1/2} c_4^{-1} \log(2^{1/2} / c_3) \log p \left( \frac{s_0^2}{n} \right)^{1/2} \) satisfies there with respect to \( \varepsilon > 0 \), that is,

\[
\sup_{0 \leq |A| \leq p_0} \int_{2^{-n/2}}^{2^{1/2}} H^{1/2}(t/c_3, f_{ji}) dt \leq p_0^{1/2} 2^{1/2} \varepsilon \log(2^{1/2} / c_3) \leq c_4 n^{1/2} \varepsilon^2. \quad (A.1)
\]

for some constant \( c_3 > 0 \) and \( c_4 > 0 \), say \( c_3 = 10 \) and \( c_4 = \frac{(2/3)^{5/2}}{512} \). By Assumption (2), \( C_{\min}(x^0) \geq \varepsilon_{n,p_0,p}^2 \) implies (A.1), provided that \( s_1^0 \geq (2c_0)^{1/2} c_4^{-1} \log(2^{1/2} / c_3) \).

92
Note that $|\mathcal{B}_i| = \binom{s^n_2}{s^n_{2-i}} \leq \frac{(|G||G| - s^n_2)^i}{i!} \leq \frac{(|G|^2/4)^i}{i!}$ by the binomial coefficients formula. Moreover, $\sum_{j=1}^{s^n_1} 2^j i^j \leq i^{s^n_1}$, and $\sum_{j_1 + \ldots + j_t = j} \binom{j}{j_1, \ldots, j_t} 2^j = (2i)^j$ using the Multinomial Theorem. By Theorem 1 of Wong and Shen (1995), there exists a constant $c_2 > 0$, say $c_2 = \frac{4}{27}$, $I$ is upper bounded by

$$I \leq \sum_{i=1}^{s^n_2} |\mathcal{B}_i| \sum_{j=1}^{s^n_1} \sum_{(j_1, \ldots, j_t)} \binom{j}{j_1, \ldots, j_t} 2^{j_1} \cdots 2^{j_t} \exp \left( - c_2 ni C_{\min}(x^0) \right)$$

$$\leq \sum_{i=1}^{s^n_2} \exp \left( - c_2 ni C_{\min}(x^0) + 2i(|G| + \log s^n_1) \right)$$

$$\leq \exp \left( - c_2 n C_{\min}(x^0) + 2(|G| + \log s^n_1) \right).$$

Let $D = \{ \hat{x} \neq \check{x}^0 \}$. For the risk property, $Eh^2(\hat{x}, x^0) = Eh^2(\hat{x}, x^0) + Eh^2(\check{x}, x^0) I(D)$ is upper bounded by

$$Eh^2(\hat{x}, x^0) + \exp \left( - c_2 n C_{\min}(x^0) + 2(|G| + \log s^n_1) \right) = (1 + o(1))Eh^2(\hat{x}, x^0),$$

using the fact that $h(\check{x}, x^0) \leq 1$. This completes the proof.
APPENDIX B

PROOF OF PROPOSITION 2
Note that under the boundedness condition, the $L_2$-norm is equivalent to the Hellinger distance under (1.17). We give the proof by verifying each assumption in Theorem 4.

We first notice that assumption 2 follows from [Kolmogorov and Tikhomirov (1959)] by plugging

$$h^2(x, x^0) = 2E \left( 1 - \exp \left( -\frac{1}{8} (Ax - Ax^0)^2 \right) \right).$$

Note that

$$\left| \frac{\partial h^2(x, x^0)}{\partial x_j} \right| \leq \frac{1}{2} E(|A_j|),$$

where $A_j$ is the $j$th column of $A$; $1 \leq j \leq p$ and $x \in \mathcal{R}^p$. Thus we can conclude

$$|h^2(x, x^0) - h^2(x_+, x^0)| = \tau \sum_{j:|x_j| \geq +} \left| \frac{\partial h^2(x, x^0)}{\partial x_j} \right|_{x=x^*} \leq 2 \tau \sum_{j:|x_j| \geq +} E(|A_j|) \leq 2\tau p \max_j \Sigma_{jj}.$$ 

Then Assumption 3 is fulfilled with $d_1 = d_2 = 1$ and $d_3 = 2 \max_j \Sigma_{jj}$.

To simplify Assumption 1, we derive an inequality through some straightforward calculations. Let $\tilde{x} = (x_{B_G}, 0) - (0, x_{B_G^0})$, where $x_{B_G}$ is obtained by removing zero components from $x$. Then

$$C_{\min}(x^0) \geq c_1^* \min_{x_{B_G} \neq x_{B_G^0}, |B_G| \leq |B_{G^0}|} |B_{G^0} \setminus B_G|^{-1} E(A_{B_G} x_{B_G} - A_{B_{G^0}} x_{B_{G^0}})^2 \geq c_1^* \min_{|B_G| \leq |B_{G^0}|, |B_G| \leq |B_{G^0}|} |B_{G^0} \setminus B_G|^{-1} \tilde{x}^T \Sigma_{B_G \cup B_{G^0}} \tilde{x} \geq \gamma_{\min}^2 \min_{B: |B| \leq 2|B_{G^0}|, B_{G^0} \subseteq B} c_{\min}(\Sigma_B).$$

for some constant $c_1^* > 0$, because the derivative of $1 - \exp(-\frac{1}{8} x^2)$ is bounded away from zero under the compactness assumption.
APPENDIX C

ACCELERATED GRADIENT METHOD
The AGM procedure is listed in Algorithms 10, in which $f(x)$ is the objective function $\frac{1}{2}\|Ax - y\|_2^2$ with $\nabla f(x)$ denotes its gradient at $x$. In addition, $f_{L,u}(x)$ is the linearization of $f(x)$ at $u$ defined as follows:

$$f_{L,u}(x) = f(u) + \nabla f(u)^T(x - u) + \frac{L}{2}\|x - u\|_2^2.$$

**Algorithm 10** Accelerated Gradient Method [Nesterov (2007); Beck and Teboulle (2009)] for (1.7)

**Input:** $A$, $y$, $s_1$, $s_2$, $L_0$, $x_0$

**Output:** solution $x$ to (1.7)

1. **Initialize:** $L_0$, $x_1 = x_0$, $\alpha_{-1} = 0$, $\alpha_0 = 1$, $t = 0$.
2. **repeat**
3. $t = t + 1$, $\beta_t = \frac{\alpha_t - 1}{\alpha_{t-1}}$, $u_t = x_t + \beta_t(x_t - x_{t-1})$
4. **Line search:** Find the smallest $L = 2^t L_{t-1}$ such that

$$f(x_{t+1}) \leq f_{L,u_t}(x_{t+1}),$$

where $x_{t+1} = \text{SGLP}(u_t - \frac{1}{L}\nabla f(u_t), s_1, s_2)$
5. $\alpha_t = \frac{1 + \sqrt{1 + 4\alpha_{t-1}^2}}{2}$, $L_t = L$.
6. **until** Convergence
7. **return** $x_t$
APPENDIX D

ALGORITHM FOR SOLVING 4.13
We give a detailed description of algorithm for solving the restricted projection (4.13) in Algorithm 11.
Algorithm 11 Restricted Sparse Group Lasso Projection Algorithm

**Input:** $v, s_1, s_2, T_1, T_3$

**Output:** an optimal solution $x$ to the Restricted Sparse Group Projection Problem (4.13)

**Function** RSGLP($v, s_1, s_2, T_1, T_3$)

1: if $\|x_{T_1}\|_1 \leq s_1$ and $\|x_{T_3}\|_G \leq s_2$ then
2: return $v$
3: end if
4: $x_{C_1} = v_{(T_1)}$, $x_{C_1} = P_{s_1}(v_{T_1})$
5: $x_{C_2} = v_{(T_3)}$, $x_{C_2} = P_{s_2}(v_{T_3})$
6: $x_{C_{12}} = v_{(T_1)}$, $x_{C_{12}} = \text{bisec}(v, s_1, s_2, T_1, T_3)$
7: if $\|x_{C_1}\|_G \leq s_2$ then
8: return $x_{C_1}$
9: else if $\|x_{C_2}\|_1 \leq s_1$ then
10: return $x_{C_2}$
11: else
12: return $x_{C_{12}}$
13: end if

**Function** bisec($v, s_1, s_2, T_1, T_3$)

1: Initialize $up$, $low$ and $tol$
2: while $up - low > tol$ do
3: $\hat{\lambda} = (low + up)/2$
4: if (4.15) has a solution $\hat{x}$ given $v\hat{\lambda}$ then
5: calculate $\hat{x}_1$ using $\hat{x}$ and $\hat{\lambda}$.
6: if $\hat{s}_1 \leq s_1$ then
7: $up = \hat{\lambda}$
8: else
9: $low = \hat{\lambda}$
10: end if
11: else
12: $up = \hat{\lambda}$
13: end if
14: end while
15: $\lambda^* = up$
16: Solve (4.14) to get $\eta^*$
17: Calculate $(x^*)_{T_1}$ from $\lambda^*$ and $\eta^*$.
18: return $(x^*)_{T_1}$
Alternating Direction Method of Multipliers (ADMM) is widely chosen for its capability of decomposing coupled variables/constraints, which is exactly the case in our projection problem. Before applying ADMM, we transform (4.8) into an equivalent form as follows:

\[
\begin{align*}
\text{minimize} \quad & \frac{1}{2} \| \mathbf{x} - \mathbf{v} \|^2_2 \\
\text{subject to} \quad & \| \mathbf{u} \|_1 \leq s_1 \\
& \| \mathbf{w} \|_G \leq s_2 \\
& \mathbf{u} = \mathbf{x}, \mathbf{w} = \mathbf{x}.
\end{align*}
\]

The augmented Lagrangian is:

\[
\mathcal{L}(\mathbf{x}, \lambda, \eta) = \frac{1}{2} \| \mathbf{x} - \mathbf{v} \|^2_2 + \lambda^T(\mathbf{u} - \mathbf{x}) + \eta^T(\mathbf{w} - \mathbf{x}) + \frac{\rho}{2}(\| \mathbf{u} - \mathbf{x} \|^2_2 + \| \mathbf{w} - \mathbf{x} \|^2_2).
\]

Utilize the scaled form \cite{Boyd2011}, i.e., let \( \lambda = \frac{\lambda}{\rho}, \eta = \frac{\eta}{\rho} \), we can obtain an equivalent augmented Lagrangian:

\[
\mathcal{L}(\mathbf{x}, \lambda, \eta) = \frac{1}{2} \| \mathbf{x} - \mathbf{v} \|^2_2 + \frac{\rho}{2}(\| \mathbf{x} - \mathbf{u} - \lambda \|^2_2 + \| \mathbf{x} - \mathbf{w} - \eta \|^2_2) - \frac{\rho}{2}(\| \lambda \|^2_2 + \| \eta \|^2_2).
\]

Now we calculate the optimal \( \mathbf{x}, \lambda \) and \( \eta \) through alternating minimization. For fixed \( \mathbf{u} \) and \( \mathbf{w} \), the optimal \( \mathbf{x} \) possesses a closed-form solution:

\[
\mathbf{x} = \frac{1}{1 + 2\rho} (\mathbf{v} + \rho(\mathbf{u} + \mathbf{\lambda} + \mathbf{w} + \mathbf{\eta})).
\]

For fixed \( \mathbf{x} \) and \( \mathbf{u} \), finding the optimal \( \mathbf{w} \) is a group lasso projection:

\[
\begin{align*}
\text{minimize}_{\mathbf{w}} \quad & \frac{1}{2} \| \mathbf{w} - (\mathbf{x} - \mathbf{\eta}) \|^2_2 \\
\text{subject to} \quad & \| \mathbf{w} \|_G \leq s_2
\end{align*}
\tag{E.1}
\]

For fixed \( \mathbf{x} \) and \( \mathbf{w} \), finding the optimal \( \mathbf{u} \) amounts to solve an \( L_1 \)-ball projection:

\[
\begin{align*}
\text{minimize}_{\mathbf{u}} \quad & \frac{1}{2} \| \mathbf{u} - (\mathbf{x} - \mathbf{\lambda}) \|^2_2 \\
\text{subject to} \quad & \| \mathbf{u} \|_1 \leq s_1
\end{align*}
\tag{E.2}
\]

The update of multipliers is standard as follows:

\[
\lambda = \lambda + \mathbf{u} - \mathbf{x} \\
\eta = \eta + \mathbf{w} - \mathbf{x}
\]

\tag{E.3}

Algorithm 12 summarizes the above procedure. Note that, the value of the penalty term \( \rho \) is fixed in Algorithm 12. However, in our implementation, we increase \( \rho \) whenever necessary to obtain faster convergence.
Algorithm 12 ADMM [Boyd et al. (2011)] for (4.8)

**Input:** \(v, s_1, s_2\)

**Output:** an optimal solution \(x\) to (4.8)

**Initialize:** \(x_0, u_0, w_0, \lambda_0, \eta_0, t = 0, \rho > 0\)

repeat
  
  \(t = t + 1\)
  
  \(x_t = \frac{1}{1+2\rho}(v + \rho(u_{t-1} + \lambda_{t-1} + w_{t-1} + \eta_{t-1}))\)

  \(w_t = P^c_G(x_t - \eta_{t-1})\)

  \(u_t = P^s_1(x_t - \lambda_{t-1})\)

  \(\lambda_t = \lambda_{t-1} + u_t - x_t, \eta_t = \eta_{t-1} + w_t - x_t.\)

until Convergence

return \(x_t\)
APPENDIX F

THE DYKSTRA’S ALGORITHM
The Dykstra’s algorithm is a general scheme to compute the projection onto intersections of convex sets. It is carried out by taking Euclidean projections onto each convex set alternatively in a smart way and is guaranteed to converge for least squares objective function \cite{Combettes2010}. The details of applying Dykstra’s Algorithm to our projection problem are listed in Algorithm 13.

\begin{algorithm}
\caption{Dykstra’s Algorithm \cite{Combettes2010} for (4.8)}
\label{alg:dykstra}
\begin{algorithmic}
\Input $v$, $s_1$, $s_2$
\Output an optimal solution $x$ to (4.8)
\Initialize $x_0 = v$, $p_0 = 0$, $q_0 = 0$, $t = 0$
\Repeat
\hspace{1em} $t = t + 1$
\hspace{1em} $y_{t-1} = P_{s_2}^G(x_{t-1} + p_{t-1})$
\hspace{1em} $p_t = x_{t-1} + p_{t-1} - y_{t-1}$
\hspace{1em} $x_t = P_{s_1}^1(y_{t-1} + q_{t-1})$
\hspace{1em} $q_t = y_{t-1} + q_{t-1} - x_t$
\Until Convergence
\Return $x_t$
\end{algorithmic}
\end{algorithm}
Proof. Let \( w^k \) denote \( x^k - \nabla f(x^k) \). It is clear that
\[
\begin{align*}
\|x^{k+1} - w^k\|_2^2 & = \|x^{k+1} - x^*\|_2^2 + \|x^* - w^k\|_2^2 + 2\langle x^{k+1} - x^* , x^* - w^k \rangle \\
& \leq \|x^* - w^k\|_2^2,
\end{align*}
\]
where the last inequality comes from the optimality of \( x^{k+1} \). After eliminating \( \|x^* - w^k\|_2^2 \) from both sides we can obtain:
\[
\begin{align*}
\|x^{k+1} - x^*\|_2 & \leq 2\langle x^{k+1} - x^* , \frac{w^k - x^*}{\|x^{k+1} - x^*\|_2} \rangle \\
= & 2\langle x^k - A^T(Ax^k - y) - x^* , \frac{x^{k+1} - x^*}{\|x^{k+1} - x^*\|_2} \rangle \\
= & 2\langle x^k - A^T(Ax^k - (Ax^* + e^*)) - x^* , \frac{x^{k+1} - x^*}{\|x^{k+1} - x^*\|_2} \rangle \\
= & 2\langle (I - A^TA)(x^k - x^*) - A^Te^* , \frac{x^{k+1} - x^*}{\|x^{k+1} - x^*\|_2} \rangle \\
= & 2\langle (I - A^TA_U)(x^k - x^*) - A^Te^* , \frac{x^{k+1} - x^*}{\|x^{k+1} - x^*\|_2} \rangle \\
\leq & 2(\|I - A^TA_U\|_2 \|x^k - x^*\|_2 + \|A\| \frac{x^{k+1} - x^*}{\|x^{k+1} - x^*\|_2} \|e^*\|_2) \\
\leq & 2(c_1 \|x^k - x^*\|_2 + \sqrt{1 + c_2} \|e^*\|_2),
\end{align*}
\]
where the set \( U \) is the union of support of \( x^* \), \( x^k \) and \( x^{k+1} \) and the last inequality is from the fact that the spectral norm of \( I - A^TA_U \) is upperbounded by \( \delta\|U\| \) [Blumensath and Davies (2009)]. The first conclusion then follows from expanding the last term and compute the power series.

To prove the second result, a finer treatment of the set \( U \) above is needed. Specifically, we consider the following four sets:
\[
\begin{align*}
I_1 & = \text{supp}(x^k), \quad I_2 = \text{supp}(x^{k+1}) \\
I_3 & = \text{supp}(x^*) - \text{supp}(x^k) \\
I_4 & = \text{supp}(x^*) - \text{supp}(x^{k+1}),
\end{align*}
\]
and it is easy to verify that:
\[
\begin{align*}
\text{supp}(x^k - x^*) & \subset I_{13} \\
\text{supp}(x^{k+1} - x^*) & \subset I_{24} \\
|I_{ij}| & = |I_i \cup I_j| \leq 2s_1, \quad \forall (i, j) \in \{1, 2, 3, 4\}.
\end{align*}
\]
Therefore we can conclude that:

\[
\begin{align*}
\|x^{k+1} - x^*\|_2 & \leq 2\|(I - A^T A)(x^k - x^*)\|_2 \\
& \leq 2\langle (I - A^T A)(x^k - x^*), (x^{k+1} - x^*) \rangle \\
& \leq 2\langle (I - A^T A)(x^k - x^*)_{I_1}, (x^{k+1} - x^*)_{I_1} \rangle \\
& \quad + 2\langle (I - A^T A)(x^k - x^*)_{I_2}, (x^{k+1} - x^*)_{I_2} \rangle \\
& \quad + 2\langle (I - A^T A)(x^k - x^*)_{I_3}, (x^{k+1} - x^*)_{I_3} \rangle \\
& \quad + 2\langle (I - A^T A)(x^k - x^*)_{I_4}, (x^{k+1} - x^*)_{I_4} \rangle \\
& \leq 2c_2||x^k - x^*||_{I_1}||x^{k+1} - x^*||_{I_2} \\
& \quad + ||x^k - x^*||_{I_2}||x^{k+1} - x^*||_{I_3} \\
& \quad + ||x^k - x^*||_{I_3}||x^{k+1} - x^*||_{I_2} \\
& \quad + ||x^k - x^*||_{I_4}||x^{k+1} - x^*||_{I_4} \\
& \leq 2c_2 \sqrt{2||x^k - x^*||_{I_1}^2 + 2||x^k - x^*||_{I_3}^2} \\
& \quad \leq 2c_2 \sqrt{2||x^{k+1} - x^*||_{I_2}^2 + 2||x^{k+1} - x^*||_{I_4}^2} \\
\end{align*}
\]

where the first inequality is from our proof of the first result and we apply the Cauchy inequality to obtain the last inequality. The proof is completed by expanding the last term and computing the resulting power series. \hfill \Box