Mining Data with Feature Interactions

by

Yashu Liu

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

Approved August 2018 by the
Graduate Supervisory Committee:

Jieping Ye, Co-chair
Guoliang Xue, Co-chair
Huan Liu
Hans D. Mittelmann

ARIZONA STATE UNIVERSITY

December 2018

ABSTRACT

Models using feature interactions have been applied successfully in many areas such as biomedical analysis, recommender systems. The popularity of using feature interactions mainly lies in (1) they are able to capture the nonlinearity of the data compared with linear effects and (2) they enjoy great interpretability. In this thesis, I propose a series of formulations using feature interactions for real world problems and develop efficient algorithms for solving them.

Specifically, I first propose to directly solve the non-convex formulation of the weak hierarchical Lasso which imposes weak hierarchy on individual features and interactions but can only be approximately solved by a convex relaxation in existing studies. I further propose to use the non-convex weak hierarchical Lasso formulation for hypothesis testing on the interaction features with hierarchical assumptions. Secondly, I propose a type of bi-linear models that take advantage of interactions of features for drug discovery problems where specific drug-drug pairs or drug-disease pairs are of interest. These models are learned by maximizing the number of positive data pairs that rank above the average score of unlabeled data pairs. Then I generalize the method to the case of using the top-ranked unlabeled data pairs for representative construction and derive an efficient algorithm for the extended formulation. Last but not least, motivated by a special form of bi-linear models, I propose a framework that enables simultaneously subgrouping data points and building specific models on the subgroups for learning on massive and heterogeneous datasets. Experiments on synthetic and real datasets are conducted to demonstrate the effectiveness or efficiency of the proposed methods.

Dedicated to my late father

ACKNOWLEDGEMENTS

I would like to first express my deepest appreciation to Professor Jieping Ye for his invaluable inspiration, guidance and support during my PhD studies at Arizona State University. I'm very grateful to Professor Ye for everything that I learned from him about thinking and solving problems and the dissertation would have been impossible without him. I would also like to extend my deepest gratitude to Professor Guoliang Xue for his endless advice, help and support and I was priviledged to have had the opportunities to work with him. I must also thank my committee members Professor Huan Liu and Professor Hans D. Mittelmann for their valuable suggestions and advice.

I would like to thank my coauthors, collaborators, labmates and friends at our labs for enlightening discussions and collaborations: Jun Liu, Shuiwang Ji, Liang Sun, Jianhui Chen, Lei Yuan, Jiayu Zhou, Zheng Wang, Pinghua Gong, Ming Lin, Jie Wang, Sen Yang, Shuo Xiang, Qian Sun, Cheng Pan, Rita Chattopadhyay, Shuang Qiu, Zhi Nie, Qingyang Li, Chao Zhang, Kefei Liu, Rashmi Dubey, Tao Yang. Special thanks to my callaborators outside ASU, Ping Zhang, Han Li, Liang Zhan, Gayle Wittenberg for their insights and feedbacks. I am very grateful to my mentor at IBM Research Kenney Ng for edifying discussions and patient guidance. Many thanks to my colleagues and friends at ASU for their care and help: Xiang Zhang, Ziming Zhao, Dejun Yang, Jin Zhang, Ruozhou Yu, Peng Wu, Lingjun Li, Xi Fang, Lu Zhang, Yangzi Liu, Wan Yu.

Finally and most importantly, I wish to thank my faimily for their unwavering support on my PhD studies. I am deeply indebted to my late father for his profound belief in me and always backing me up. I would like to thank my wife Yahan Chen, father-in-law Yue Chen and mother-in-law Fulan Zhai for their unconditioned support and understanding. I am very grateful to my uncle Jiaqi Liu and cousin Yajun Liu for helping me pull through the difficult times.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

Table Page

Table Page

LIST OF FIGURES

Chapter 1

BACKGROUND AND INTRODUCTION

1.1 Basics for Models Using Feature Interactions

Feature interactions have been widely used and studied in communities of statistics, machine learning and data mining. However, the concept of interaction can be ambiguous and general. Existing study (Jakulin, 2005) categorizes the meanings of interaction to two senses: 1) ontic sense which refers to an ambiguous or bidirectional causal relationship and 2) epistemic sense which refers to a type of association, correlation or entanglement. This thesis focuses on the interactions in epistemic sense.

An feature interaction may involve multiple features and it can be flexibly defined in mathematical form. A k-way interaction for k features $x_1, x_2, \ldots, x_k$ can be defined by a function $\mathfrak{I} : \mathbb{R}^k \to \mathbb{R}$:

$$x_{1,2,\ldots,k} = \mathfrak{I}(x_1, x_2, \ldots, x_k). \tag{1.1}$$

The most commonly used operator to define feature interaction is the multiplication operator. For example, the interaction of k features can be written as $x_1 \cdot x_2 \cdot \ldots \cdot x_k$, which can be viewed as a k-th order monomial term. I mainly focus on the interactions defined by multiplicative operator in this thesis.

Consider a linear regression model with the outcome variable y and d features $x_1, \ldots, x_d$:

$$y = w_0 + \sum_{i=1}^{d} x_i w_i + \epsilon, \tag{1.2}$$

where w_0 is the bias term, $w_i, i = 1 \ldots, d$ is the coefficient and $\epsilon \sim N(0, \sigma^2)$ is the noise term. In many real-world applications, a simple linear regression model is

not sufficient for predictive or explanatory purposes. One strategy to capture the nonlinearity of the data which has recently received increasing attention in statistics and machine learning community is to include higher-order interaction terms into the model (Hastie *et al.*, 2009; Montgomery *et al.*, 2012). For example, the regression model including terms of order-2 and lower has the following form:

$$y = w_0 + \sum_{i=1}^{d} x_i w_i + \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d} x_i x_j W_{i,j} + \epsilon, \tag{1.3}$$

where the cross-product term $x_i x_j$, $i \neq j$ refers to as the 2-way interaction variable (one may view x_i^2 as a special interaction variable), and $w_i's$ and $W \in \mathbb{R}^{d\times d}$ are called the main effect and interaction effect coefficients respectively. In matrix form, equation (1.3) can be written as:

$$y = w_0 + \mathbf{w}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T W\mathbf{x} + \epsilon, \tag{1.4}$$

where $\mathbf{x} = [x_1, x_2, \ldots, x_d]^T \in \mathbb{R}^d$, $\mathbf{w} = [w_1, w_2, \ldots, w_d]^T \in \mathbb{R}^d$ and the intercept term w_0 is ommited here for notational simplicity. Besides the benefits of increasing the complexity of a linear model, an interaction term enjoys great interpretability. That is, an interaction $x_i \cdot x_j$ represents the effect that is produced by changing one feature (say x_i) depends on the level of the other feature (x_j) (Montgomery *et al.*, 2012).

A bi-linear regression model can be viewed as a general model for two-way interactions of which the formulation is written as

$$y = w_0 + \sum_{i=1}^{d} x_i w_i + \sum_{j=1}^{p} z_i u_i + \sum_{i=1}^{d}\sum_{j=1}^{p} x_i z_j W_{i,j} + \epsilon, \tag{1.5}$$

where coefficient w_i, u_j model the linear terms x_i, z_j, $W_{i,j}$ models the interaction $x_i \cdot z_j$ and ϵ is the error term. Analogeously, one may write (1.5) in a matrix form:

$$y = w_0 + \mathbf{w}^T\mathbf{x} + \mathbf{u}^T\mathbf{z} + \mathbf{x}^T W\mathbf{z} + \epsilon, \tag{1.6}$$

where y is the outcome variable, $\mathbf{x} = [x_1, x_2, \ldots, x_d]^T \in \mathbb{R}^d$ and $\mathbf{z} = [z_1, z_2, \ldots, z_p]^T \in \mathbb{R}^p$ are the feature vectors, $\mathbf{w} = [w_1, w_2, \ldots, w_d]^T \in \mathbb{R}^d$ and $\mathbf{u} = [u_1, u_2, \ldots, u_p]^T \in \mathbb{R}^p$ are coefficient vectors. It is straightforward that model (1.5) can be degenerated to (1.3)with $\mathbf{z}$ replacd by $\mathbf{x}$.

Applications with interaction regression models are omnipresent. For example, in psychological study, the effectiveness of using 3-way interactions was demonstrated in testing psychological hypothesis (Dawson and Richter, 2006); there are strong evidences found that genetic-environment interactions have significant effects on conduct disorders (Cadoret *et al.*, 1995); Eley *et al.* (2004) found a couple of evidences of gene-environment interactions in predicting depression status; the interaction between continuance commitment and affective commitment was found significant in predicting job withdraw intentions and absenteeism (Somers, 1995); Gatt *et al.* (2009) discovered that brain-derived neurotrophic factor interacts with early life stress in predicting cognitive features of depression and anxiety.

1.2 Hierarchical Structures for Modeling Feature Interactions

The use of higher order interaction terms leads to data of high dimensionality. For instance, for regression model (1.2), if one wants to add all terms of order-k and lower, then there will be a total of $\mathcal{O}(d^k)$ variables, which is computationally demanding for parameter estimation even when k and d are fairly small. Thus, an efficient approach that is able to deal with huge dimensionality is desired in such cases, and the sparse learning methodology is one promising approach for tackling such problem (Tibshirani, 1996; Koh *et al.*, 2007; d'Aspremont *et al.*, 2004; Candes and Romberg, 2006; Zou *et al.*, 2006).

In general, not all of the main effects and interactions are of interest, thus it is critical to select the variables of great significance. One simple approach for high

dimensional interaction regression is to directly apply the Lasso (Tibshirani, 1996). In the case of 2-way interactions, the "all-pairs Lasso" (Bien *et al.*, 2013) optimizes the following objective:

$$\min_{w_0,\mathbf{w},W} \frac{1}{2n} \sum_{i=1}^{n} \|y_i - w_0 - \mathbf{w}^T\mathbf{x}_i + \frac{1}{2}\mathbf{x}_i^T W \mathbf{x}_i\|_2^2 + \lambda\|\mathbf{w}\|_1 + \lambda\|W\|_1, \tag{1.7}$$

where n is the sample size, $\|\mathbf{w}\|_1 = \sum_i |w_i|$, $\|W\|_1 = \sum_{i,j} |W_{i,j}|$ and λ is the penalty parameter. The l_1-norm pernalties (also known as the Lasso penalties (Tibshirani, 1996)) are well known to result in sparse solutions to the coefficients. However, the all-pairs Lasso estimator does not account for any hierarchical structural information between main effects and interactions which has been shown to be very effective in constraining the search space and identifying important individual features and interactions (Bien *et al.*, 2013; Zhao *et al.*, 2009; Radchenko and James, 2010; Yuan *et al.*, 2009; Choi *et al.*, 2010). Specifically, the hierarchical constraint requires that an interaction term $x_i x_j$ is selected in the model only if the main (parent) effects x_i and/or x_j are included. The herarchical structures are usually categorized into two types (Chipman, 1996). The *strong hierarchy* requires that the interaction effects are non-zero only if the corresponding main effects are non-zero. In the example of two-way interactions, strong hierarchy indicates that $W_{i,j} \neq 0$ only if $w_i \neq 0$ AND $w_j \neq 0$. Different from the strong hierarchy, the *weak hierarchy* between the main effects and the interaction effects requires that an interaction is included in the model only if at least one of the main effects is included in the model, *i.e.,* $W_{i,j} \neq 0$ only if $w_i \neq 0$ OR $w_j \neq 0$. The weak hierarchy can be considered as a structure in between the strong hierarchy and no hierarchical structure (Bien *et al.*, 2013; Yuan *et al.*, 2009; Zhao *et al.*, 2009). Specifically, weak hierarchy allows those interactions with only one significant "parent" (main effect) to be included in the model. Strong theoretical properties have been established for such hierarchical model (Yuan *et al.*, 2009; Zhao

et al., 2009). The hierarchical structure is supported by the argument that large main effects may result in interactions of more importance, and it is desired in a wide range of applications in engineering and underlying science.

Traditional approaches to fit such a model typically follow the following two-step procedures (Montgomery *et al.*, 2012):

- Fit a linear regression model that only includes the main effects and then select the significant features;
- Fit the reformulated model with the identified individual features and the interactions constructed via domain knowledge.

Since even a small d may lead to a huge amount of interaction variables, the two-step procedure is still time-consuming in many applications. Recently, there have been growing research efforts on imposing the hierarchical structure on main effects and interactions in the regression model with novel sparse learning methods.

Yuan *et al.* (2009) proposed a type of non-negative garrote method to achieve the strong and weak hierarchical structures by imposing constraints

$$w_i \geq 0, \quad W_{i,j} \geq 0, \quad W_{i,j} \leq \min(w_i, w_j)$$

and

$$w_i \geq 0, \quad W_{i,j} \geq 0, \quad W_{i,j} \leq w_i + w_j$$

to the regression objective respectively. Zhao *et al.* (2009) proposed the Composite Absolute Penalties (CAP) family which take advantege of the properties of norm penalties at overlapping and non-overlapping groups to impose heredity structures for interaction models. The core principle of hierarchical CAP is to penalize both the groups of the desendents effects without their parents and the groups the descendent effects and their parents appear together. The former non-overlapping group penalty

enables variable selection while the latter overlapping group penalty achieves the hierarchical structures. In the example of 2-way interactions, the CAP uses the following overlapping group pattern to achieve the hierarchichal structures:

$$\lambda \sum_{i \neq j} \left(|W_{i,j}| + \| [w_i, w_j, W_{i,j}] \| \right) \|_\gamma,$$

where $\gamma > 1$, λ controls the penalty amount. Motivated by closing ideas, Radchenko and James (2010) invented the VANISH algorithm which adopts a analogous principle to penalize nested groups

$$\sum_{j=1}^{d} \left(\lambda_1 \| \left[w_j, W_{\cdot,j} \right] \|_2 + \lambda_2 \| W_{\cdot,j} \|_1 \right),$$

to achieve the hierarchical structure between the interaction effects and main effects. Recently, Bien *et al.* (2013) have proposed the strong hierarchical Lasso to achieve both strong heredity structural solutions and simultaneous feature selection which adds a set of constraints, *i.e.,*

$$W = W^T, \quad \|W_{\cdot,j}\|_1 \leq |w_j|, \quad j = 1, \ldots, d,$$

to the all pairs Lasso formulation (1.7). Meanwhile, they remove the symmetric constraints to obtain a solution with the weak hierarhical structure. The hierarchical constraints can be equally expressed as a penalty in the form of

$$\lambda \sum_j \left(\max(|w_j|, \|W_{\cdot,j}\|_1) + \frac{1}{2} \|W_{\cdot,j}\|_1 \| \right)$$

which is closing to the spirit of CAP and VANISH. In contrast to the above works which fulfill the hierarchical structure via solving convex problems, Choi *et al.* (2010) proposed a non-convex formulation for strong hierarchy by modeling the coefficient of an interaction term is a product of a scalar and main effect coefficients, i.e.,

$$W_{i,j} = \gamma_{i,j} w_i w_j,$$

where $\gamma_{i,j}$ is newly introduced coefficient. The Lasso penalty is applied on both coeffciients $\gamma_{i,j}$'s and $\mathbf{w}$ which results in strong hierarchical structures. The non-convex formulation is solved via alternating update strategies.

1.3 Drug Discovery Problems - From Data Pair Interactions to Feature Interactions

Drug discovery is a time-consuming and laborious process. By conservative estimates, it now takes at least 10 to 15 years and $500 million to $2 billion to bring a single drug to market (Adams and Brantner, 2006). Furthermore, there is a widening productivity gap: research and development spending continues to increase, yet the number of new therapeutic chemical and biological entities approved by the US FDA has been declining since the late 1990s. The lack of efficacy (i.e., whether the drug works better than alternatives) and safety issues (i.e., whether the drug brings serious adverse event and/or drug-drug interactions) are the two major reasons for which a drug fails clinical trials, each accounting for around 30% of failures (Hopkins, 2008). Thus the development of computational techniques to predict drug effects and drug-drug interactions holds great promise for reducing the attrition rate and improving the drug discovery process.

Drug repositioning is the process of finding additional indications (i.e., diseases) for existing drugs. At the same time, as the number of approved drugs is continuously increasing, Drug-Drug Interaction (DDI) has become a serious health and safety issue which draws great attention from both academia and industry. Numerous methods have also been developed for predicting potential DDIs in the last decade (Iyer *et al.*, 2014; Tatonetti *et al.*, 2011; Luo *et al.*, 2014).

Both drug repositioning and DDI prediction can be regarded as a binary dyadic prediction problem, which aims to predict the "label" of a data pair. For the drug repositioning problem, a data pair would be composed of a drug and a disease, and its

label is $+1$ if the drug can treat the disease, and -1 otherwise. For DDI prediction, a data pair includes two different drugs, and its label is $+1$ if there is an interaction between the two drugs, and -1 otherwise. In both problems, the most general setting is that we have a small set of positive (*i.e.,* $+1$ labeled) data pairs, while the labels of the remaining data pairs are unknown. Most of the existing computation based methodologies treat those unknown data pairs' labels in analogous applications as -1 (Natarajan and Dhillon, 2014; Gonen and Kaski, 2014).

For the dyadic prediction problem, the learning task is to find a function $f : \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \to \mathbb{R}$ for a data pair $(\mathbf{x}, \mathbf{z}) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2}$:

$$y = f(\mathbf{x}, \mathbf{z}), \tag{1.8}$$

where y is the label to be predicted. The common choice for function f is the bilinear function, *i.e.,*

$$f(\mathbf{x}, \mathbf{z}) = \mathbf{x}^T W \mathbf{z}, \tag{1.9}$$

where $W \in \mathbb{R}^{d_1 \times d_2}$ is the coeffcieint matrix. Many real-world applications achieve successful prediciton performances by adopting the bilinear models. Natarajan and Dhillon (2014) introduced an inductive matrix completion method (Jain and Dhillon, 2013) to predict the associations of gene-disease pair using a bilinear model where the coefficient matrix is fatorized as

$$W = UH^T \tag{1.10}$$

and the unknown U, H are solved via alternating minimization. Analogously, Yan *et al.* (2014) proposed to modeling the click through rate (CTR) problem with a bilinear model where the user feature vector and the advertisement feature vector consist of a data pair. The coefficient matrix W was also factorized as (1.10) but penalized with group-Lasso penalties. One may observe that the bilinear model es-

sentially makes predictions of data pairs $(\mathbf{x}, \mathbf{z})$ using their feature interactions $x_i z_j$'s based on (1.5).

Chapter 2

THE WEAK HIERARCHICAL LASSO

2.1 Introduction

Our previous empirical studies have demonstrated the stronger predictive power of weak hierarchical model in biomedical applications (Li *et al.*, 2014a). By imposing restrictions of the weak hierarchy and taking advantage of the Lasso penalty (Tibshirani, 1996) that leads to sparse coefficients, the weak hierarchical Lasso is able to simultaneously attain a hierarchical solution and identify important main effects and interactions. However, the set of constraints restricting hierarchical structures make the problem non-convex; the algorithm proposed by (Bien *et al.*, 2013) aims to solve a convex relaxation. The convex relaxation, however, requires additional conditions to guarantee the weak hierarchy, which is not desirable.

In this thesis, we propose to directly solve the weak hierarchical Lasso using the GIST (General Iterative Shrinkage and Thresholding) optimization framework recently proposed by (Gong *et al.*, 2013a). The GIST framework has been shown to be highly efficient for solving large-scale non-convex problems. The most critical step in GIST is to compute a sequence of proximal operators (Parikh and Boyd, 2013). We first show that the proximal operator related to weak hierarchical Lasso admits an analytical form solution by factorizing unknown coefficients into sign matrices and non-negative coefficients. However, a naive method of computing the subproblem of the proximal operator leads to a quadratic time complexity, which is not desirable for large-size problems. To this end, we further develop an efficient algorithm for solving the subproblems, which achieves a linearithmic time complexity. We evaluate the

efficiency and effectiveness of the proposed algorithm and compare it with the convex relaxation (Bien *et al.*, 2013) and other state-of-the-art methods using synthetic and real data sets. Our empirical study demonstrates the high efficiency of our algorithm and the superior predictive performance of weak hierarchical Lasso over the competing methods.

Furthermore, we propose to directly use the non-convex formulation for hierarchical testing of interactions (Bien *et al.*, 2015). Significance testing of interactions has always been an important but challenging problem in statistics. Starting from "backward model", Simon and Tibshirani (2012) proposed a permutation-based method, called TMIcor, for the testing of pairwise interactions for binary classification problems. In particular, an interaction between feature i and feature j is tested as significant if the absolute difference of the Fisher transformed sample correlation between two classes is greater than a threshold. The test statistics in TMIcor can be modeled as the largest Lasso penalty resulting in nonzero coefficients. In order to incorporate structural information to the testing of interactions, Bien *et al.* (2015) proposed the convex hierarchical testing framework which adopts the convex relaxation of weak hierarchical Lasso. In this thesis, instead of using convex relaxation, we propose to directly use the non-convex formulation for hierarchical testing of interactions and show the test statistics in this framework admit closed form solutions. We conduct simulation studies to compare the non-convex formulation with the convex relaxation and the results show the superiority of the non-convex formulation when a weak hierarchical structure exists in the data.

The remaining of this chapter is organized as follows: we give a brief review of the weak hierarchical Lasso and its convex relaxation. Then we derive the closed form solution to the proximal operator of the original weak hierarchical Lasso by decomposing the unknown coefficients into signs and the non-negative coefficients.

We next show how the associated proximal operator can be computed efficiently. We introduce the non-convex formulation for hierarchical testing of interactions and show the closed form solutions to corresponding test statistics and report the experimental results at the end.

2.2 The Weak Hierarchical Lasso

In this section, we briefly review the weak hierarchical Lasso and its corresponding convex relaxed formulation (Bien *et al.*, 2013). Suppose we are given n pairs of data points $\{(\mathbf{x}_i, y_i)\}_{i=1}^n \subset \mathbb{R}^d \times \mathbb{R}$. Let $\mathbf{y} \in \mathbb{R}^n$ be the vector of outcome and $X \in \mathbb{R}^{n \times d}$ be the design matrix. Let $Z \in \mathbb{R}^{n \times (d \cdot d)}$ be the matrix of interactions where

$$Z = \left[Z^{(1)}, Z^{(2)}, \ldots, Z^{(d)}\right],$$

$Z^{(i)} \in \mathbb{R}^{n \times d}$ and each column of $Z^{(i)}, i = 1, \ldots d$ is an interaction, *i.e.,* $Z^{(i)}_{\cdot,j} = X_{\cdot,i} \circ X_{\cdot,j}$ ($\circ$ is the operator of element-wise product). Thus, $Z^{(i)}$ captures the pairwise interactions between the i-th feature and all d features. Note that, we include the quadratic terms x_i^2 in the interaction model for clearer presentation, however our analysis is still applicable if they are not included in the model. By assuming that $\mathbf{y}$ is centered and X, Z are column-wise normalized to zero mean and unit standard deviation, we can set the bias term $w_0 = 0$. Thus, in matrix form, the pairwise interaction regression model can be expressed as

$$\mathbf{y} = X\mathbf{w} + \frac{1}{2} Z \cdot \mathrm{vec}(W) + \epsilon, \tag{2.1}$$

where $\epsilon \sim N(\mathbf{0}, \sigma^2 \mathbf{I})$ and "vec" is the vectorization operator that transforms a matrix to a column vector by stacking the columns of the matrix. Thus, the least square loss function of (2.1) is given by:

$$\ell(\mathbf{w}, W) = \frac{1}{2} \left\| \mathbf{y} - X\mathbf{w} - \frac{1}{2} Z \cdot \mathrm{vec}(W) \right\|_2^2. \tag{2.2}$$

Then, the weak hierarchical Lasso formulation takes the form of (Bien *et al.*, 2013):

$$\begin{aligned} &\min_{\mathbf{w},W} \ \ell\left(\mathbf{w}, W\right) + \lambda\|\mathbf{w}\|_1 + \frac{\lambda}{2}\|W\|_1 \\ &\text{s.t. } \|W_{\cdot,j}\|_1 \leq |w_j| \quad \text{for} \quad j = 1, \ldots, d, \end{aligned} \tag{2.3}$$

where $\|W\|_1 = \sum_{i,j} |W_{i,j}|$ and λ is the Lasso penalty parameter.

Note that the constraints in (2.3) guarantee the weak hierarchical structure since the coefficient $W_{i,j}$ of interaction $x_i x_j$ is non-zero only if at least one of its main effects is included in the model, *i.e.,* $w_i \neq 0$ or $w_j \neq 0$. However, the imposed hierarchical constraints make problem (2.3) non-convex. Instead of solving (2.3), Bien *et al.* (2013) proposed to solve the following relaxed version:

$$\begin{aligned} \min_{\mathbf{w}^+,\mathbf{w}^-,W} \ & \ell\left(\mathbf{w}^+ - \mathbf{w}^-, W\right) + \lambda \mathbf{1}^T(\mathbf{w}^+ + \mathbf{w}^-) + \frac{\lambda}{2}\|W\|_1 \\ \text{s.t.} \quad & \left.\begin{aligned} &\|W_{\cdot,j}\|_1 \leq w_j^+ + w_j^- \\ &w_j^+ \geq 0 \\ &w_j^- \geq 0 \end{aligned}\right\} \quad \text{for} \quad j = 1, \ldots, d, \end{aligned} \tag{2.4}$$

where $\mathbf{1}$ represents a column vector of all ones. In view of (2.4), we can see that $\|\mathbf{w}\|_1$ is relaxed to $\mathbf{w}^+ + \mathbf{w}^-$. Problem (2.4) is convex and can be solved by many efficient solvers such as FISTA (Beck and Teboulle, 2009). However, Bien *et al.* (2013) showed that problem (2.4) needs an additional ridge penalty to guarantee the weak hierarchical structure of the estimator. In this article, we propose an efficient algorithm which directly solves the non-convex weak hierarchical Lasso formulation in (2.3).

2.3 The Proposed Algorithm

In this section, we propose an efficient algorithm named "eWHL", which stands for "**e**fficient **W**eak **H**ierarchical **L**asso", to directly solve the weak hierarchical Lasso.

eWHL makes use of the optimization framework of GIST (General Iterative Shrinkage and Thresholding) due to its high efficiency and effectiveness for solving non-convex sparse formulations. One of the critical steps in GIST is to compute the proximal operator associated with the penalty functions. As one of our major contributions, we first factorize the unknown coefficients into the product of their signs and magnitudes; and then show that the proximal operator of (2.3) admits a closed form solution. We further present an efficient algorithm for computing the proximal operator associated with the non-convex weak hierarchical Lasso. The time complexity of solving each subproblem of the proximal operator can be reduced from quadratic to linearithmic.

2.3.1 The Closed Form Solution to the Proximal Operator

In this section, we show how to derive the closed form solution to the proximal operator associated with (2.3) in detail. Let

$$\mathcal{P} = \Big\{ (\mathbf{w}, W)\,,\ \mathbf{w} \in \mathbb{R}^d,\ W \in \mathbb{R}^{d\times d} \ \Big|\ \|W_{\cdot,j}\|_1 \le |w_j|,\ j = 1, \ldots, d \Big\}$$

and the indicator function be defined by

$$\mathcal{R}(\mathbf{w}, W) = \begin{cases} \lambda\|\mathbf{w}\|_1 + \dfrac{\lambda}{2}\|W\|_1\,, & \text{if} \quad (\mathbf{w}, W) \in \mathcal{P} \\ +\infty, & \text{if} \quad (\mathbf{w}, W) \notin \mathcal{P} \end{cases}. \tag{2.5}$$

Thus, problem (2.3) can be solved by iteratively generating a sequence $\left\{\mathbf{w}^{(k)}, W^{(k)}\right\}$ by:

$$\begin{aligned} \left(\mathbf{w}^{(k+1)}, W^{(k+1)}\right) = \underset{\mathbf{w},\, W}{\arg\min}\ & \ell\left(\mathbf{w}^{(k)}, W^{(k)}\right) + \left\langle \nabla_{\mathbf{w}} \ell\left(\mathbf{w}^{(k)}, W^{(k)}\right),\ \mathbf{w} - \mathbf{w}^{(k)} \right\rangle \\ & + \left\langle \nabla_{W} \ell\left(\mathbf{w}^{(k)}, W^{(k)}\right),\ W - W^{(k)} \right\rangle + \frac{t^{(k)}}{2}\left\|\mathbf{w} - \mathbf{w}^{(k)}\right\|_2^2 \\ & + \frac{t^{(k)}}{2}\left\|W - W^{(k)}\right\|_F^2 + \mathcal{R}(\mathbf{w}, W), \end{aligned} \tag{2.6}$$

where $t^{(k)} > 0$.

Simple algebraic manipulation leads to

$$\left(\mathbf{w}^{(k+1)}, W^{(k+1)}\right) = \underset{\mathbf{w},\, W}{\arg\min}\ \frac{1}{2}\left\|\mathbf{w} - \mathbf{v}^{(k)}\right\|_2^2 + \frac{1}{2}\left\|W - U^{(k)}\right\|_2^2 + \frac{1}{t^{(k)}}\mathcal{R}(\mathbf{w}, W), \quad (2.7)$$

where

$$\begin{aligned}\mathbf{v}^{(k)} =& \mathbf{w}^{(k)} - \nabla_{\mathbf{w}}\mathcal{L}\left(\mathbf{w}^{(k)}, W^{(k)}\right)/t^{(k)},\\ U^{(k)} =& U^{(k)} - \nabla_{W}\mathcal{L}\left(\mathbf{w}^{(k)}, W^{(k)}\right)/t^{(k)}.\end{aligned}$$

Problem (2.7) is the proximal operator problem associated with weak hierarchical Lasso. Because $\mathcal{R}(\mathbf{w}, W)$ is an indicator function, we can rewrite the proximal operator (2.7) as

$$\begin{aligned}\underset{\mathbf{w},\, W}{\arg\min}\ & \frac{1}{2}\left\|\mathbf{w} - \mathbf{v}\right\|_2^2 + \frac{1}{2}\left\|W - U\right\|_F^2 + \frac{\lambda}{t}\|\mathbf{w}\|_1 + \frac{\lambda}{2t}\left\|W\right\|_1\\ \text{s.t. }& \left\|W_{\cdot,j}\right\|_1 \le |w_j| \quad \text{for} \quad j = 1, \ldots, d.\end{aligned} \quad (2.8)$$

We omit the superscripts for notational simplicity.

The vector of main effect coefficients can be written as

$$\mathbf{w} = \mathbf{s}^{(0)} \circ \widetilde{\mathbf{w}},$$

where $\widetilde{w}_j = |w_j|, j = 1, \ldots, d$ and $\mathbf{s}^{(0)} \in \mathbb{R}^d$ is a column vector whose j-th element is the sign of w_j, *i.e.*, $s_j^{(0)} = \text{sign}(w_j)$. We define

$$\text{sign}(w) = \begin{cases} 1 & \text{if} \quad w > 0\\ -1 & \text{if} \quad w < 0\\ 0 & \text{if} \quad w = 0 \end{cases}, \quad (2.9)$$

and we assume in this article that the sign operator is applied on vectors or matrices elementwise. Similarly, we factorize each column of the interaction coefficient matrix

as $W_{\cdot,j} = \mathbf{s}^{(j)} \circ \widetilde{W}_{\cdot,j},\ j = 1 \ldots, d$, where $\widetilde{W}_{i,j} = |W_{i,j}|$ and $\mathbf{s}^{(j)} \in \mathbb{R}^d$ is the sign vector. Then, the proximal operator (2.8) is equivalent to

$$\begin{aligned} &\underset{\mathbf{w},\, W}{\arg\min}\ \frac{1}{2}\left\|\mathbf{w}-\mathbf{v}\right\|_2^2 + \frac{1}{2}\left\|W-U\right\|_F^2 + \frac{\lambda}{t}\|\mathbf{w}\|_1 + \frac{\lambda}{2t}\left\|W\right\|_1 \\ &\left.\begin{aligned} \text{s.t.}\ \ &\|W_{\cdot,j}\|_1 \le |w_j| \\ &w_j = \mathbf{s}_j^{(0)} \circ \widetilde{w}_j \\ &W_{\cdot,j} = \mathbf{s}^{(j)} \circ \widetilde{W}_{\cdot,j} \\ &\widetilde{w}_j \ge 0 \\ &\widetilde{W}_{\cdot,j} \succeq \mathbf{0} \end{aligned}\right\} \quad \text{for} \quad j = 1, \ldots, d, \end{aligned} \tag{2.10}$$

where $\widetilde{W}$, $\widetilde{\mathbf{w}}$ and $\mathbf{s}^{(j)},\ j = 0, \ldots, d$ are the unknown variables, $\succeq$ is defined as the element-wise "greater than or equal to" comparison operator, *i.e.,* for $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d, \mathbf{a} \succeq \mathbf{b} \Leftrightarrow a_i \ge b_i, i = 1 \ldots, d$. Therefore, the solutions to the original weak hierarchical Lasso can be obtained by iteratively solving (2.10). Note that the amounts of l_1 penalties on $\mathbf{w}$ and W can be different. Here we use the same penalty parameter λ for notational simplicity and consistency with the original formulation of weak hierarchical Lasso (2.3) studied in (Bien *et al.*, 2013). Though the factorization introduces more variables and constraints, we show that the resulting proximal operator admits a closed form solution. More importantly, we show that each sub-problem of the proximal operator can be solved by the proposed eWHL algorithm in linearithmic time. Indeed, the factorization of $\mathbf{w}$ and W into their signs and magnitudes is the first key to directly solve the original weak hierarchical Lasso.

The proximal operator in (2.10) can be decoupled into d subproblems as follows:

$$\begin{aligned} &\underset{\widetilde{w}_j, s_j^{(0)}, \widetilde{W}_{\cdot,j}, \mathbf{s}^{(j)}}{\arg\min}\ \frac{1}{2}\left(s_j^{(0)}\widetilde{w}_j - v_j\right)^2 + \frac{1}{2}\left\|\mathbf{s}^{(j)} \circ \widetilde{W}_{\cdot,j} - U_{\cdot,j}\right\|_2^2 + \frac{\lambda}{t}\widetilde{w}_j + \frac{\lambda}{2t}\mathbf{1}^T\widetilde{W}_{\cdot,j} \\ &\left.\begin{aligned} \text{s.t.}\ \ &\mathbf{1}^T\widetilde{W}_{\cdot,j} \le \widetilde{w}_j \\ &\widetilde{W}_{\cdot,j} \succeq \mathbf{0} \end{aligned}\right\}, \quad \text{for } j = 1, \ldots, d. \end{aligned} \tag{2.11}$$

Next, we show that (2.11) has a closed form solution. Since

$$\frac{1}{2}\left(w_j - v_j\right)^2 = \frac{1}{2}\left(s_j^{(0)}\widetilde{w}_j - v_j\right)^2 = \frac{1}{2}\left(s_j^{(0)}\left(s_j^{(0)}\widetilde{w}_j - v_j\right)\right)^2 = \frac{1}{2}\left(\widetilde{w}_j - s_j^{(0)}v_j\right)^2$$

and $\widetilde{w}_j \geq 0$, $s_j^{(0)}$ must have the same sign as v_j, that is, w_j has the same sign as v_j. Otherwise, the value of $\frac{1}{2}\left(\widetilde{w}_j - s_j^{(0)}v_j\right)^2$ will not achieve the minimum. Similarly, one can show that $s_i^{(j)}$, *i.e.,* the sign of $W_{i,j}$, must be the same as the sign of $U_{i,j}$. Thus, we have $\mathbf{s}^{(0)} = \text{sign}(v)$ and $\mathbf{s}^{(j)} = \text{sign}\left(U_{\cdot,j}\right)$ for $j = 1, \ldots, d$. Next, we show how to compute $\widetilde{\mathbf{w}}$ and $\widetilde{W}$.

By letting $\widetilde{v}_j = s_j^{(0)}v_j$ and $\widetilde{U}_{\cdot,j} = \mathbf{s}^{(j)} \circ U_{\cdot,j}$, each subproblem (2.11) is equivalent to

$$\begin{aligned} \underset{\widetilde{w}_j, \widetilde{W}_{\cdot,j}}{\arg\min}\ & \frac{1}{2}\left(\widetilde{w}_j - \widetilde{v}_j\right)^2 + \frac{1}{2}\left\|\widetilde{W}_{\cdot,j} - \widetilde{U}_{\cdot,j}\right\|_2^2 + \frac{\lambda}{t}\widetilde{w}_j + \frac{\lambda}{2t}\mathbf{1}^T\widetilde{W}_{\cdot,j} \\ \text{s.t.}\ & \mathbf{1}^T\widetilde{W}_{\cdot,j} \leq \widetilde{w}_j \\ & \widetilde{W}_{\cdot,j} \succeq \mathbf{0} \end{aligned}. \tag{2.12}$$

It can be verified that solving problem (2.12) is equivalent to:

$$\begin{aligned} \min_{\widetilde{w}_j, \widetilde{W}_{\cdot,j}}\ & \frac{1}{2}\left(\widetilde{w}_j - \breve{v}_j\right)^2 + \frac{1}{2}\left\|\widetilde{W}_{\cdot,j} - \breve{U}_{\cdot,j}\right\|_2^2 \\ \text{s.t.}\ & \mathbf{1}^T\widetilde{W}_{\cdot,j} \leq \widetilde{w}_j \\ & \widetilde{W}_{\cdot,j} \succeq \mathbf{0} \end{aligned}, \tag{2.13}$$

where $\breve{v}_j = \widetilde{v}_j - \frac{\lambda}{t}\mathbf{1}$ and $\breve{U}_{\cdot,j} = \widetilde{U}_{\cdot,j} - \frac{\lambda}{2t}\mathbf{1}$.

We solve (2.13) by deriving its dual problem. Let $\gamma \geq 0$ be the Lagrangian multiplier dual variable of the first inequality constraint. Define the Lagrangian function of (2.13) as:

$$\mathcal{L}(\gamma, \widetilde{w}, \widetilde{\mathbf{q}}) = \frac{1}{2}\left(\widetilde{w} - \breve{v}\right)^2 + \frac{1}{2}\left\|\widetilde{\mathbf{q}} - \breve{\mathbf{u}}\right\|_2^2 + \gamma\left(\mathbf{1}^T\widetilde{\mathbf{q}} - \widetilde{w}\right)$$

where we write $\widetilde{\mathbf{q}}$ for $\widetilde{W}_{\cdot,j}$ and $\breve{\mathbf{u}}$ for $\breve{U}_{\cdot,j}$ and omit the subscripts on $\widetilde{w}$ and $\widetilde{v}$ for simplicity with slight abuse of notation. Since the constraint $\mathbf{1}^T\widetilde{\mathbf{q}} \leq \widetilde{w}$ is affine, the

strong duality holds for the minimization problem (2.13). Thus, the dual problem of (2.13) is:

$$\max_{\gamma \geq 0} \min_{\widetilde{w}, \widetilde{\mathbf{q}} \succeq \mathbf{0}} \frac{1}{2} \left(\widetilde{w} - \breve{v}\right)^2 + \frac{1}{2} \left\|\widetilde{\mathbf{q}} - \breve{\mathbf{u}}\right\|_2^2 + \gamma \left(\mathbf{1}^T \widetilde{\mathbf{q}} - \widetilde{w}\right). \tag{2.14}$$

By rearranging the terms, (2.14) is equivalent to:

$$\max_{\gamma \geq 0} \min_{\widetilde{w}, \widetilde{W} \succeq \mathbf{0}} \frac{1}{2} \left(\widetilde{w} - (\breve{v} + \gamma)\right)^2 + \frac{1}{2} \left\|\widetilde{\mathbf{q}} - (\breve{\mathbf{u}} - \gamma \mathbf{1})\right\|_2^2 + h(\gamma), \tag{2.15}$$

where

$$h(\gamma) = -\breve{v}\gamma - \frac{1}{2}\gamma^2 + \gamma \mathbf{1}^T \breve{\mathbf{u}} - \frac{1}{2}\gamma^2 \mathbf{1}^T \mathbf{1}.$$

For fixed γ, in order to obtain the minimum of the objective function in (2.15), we conclude that

$$\begin{aligned} \widetilde{w} &= \max(\breve{v} + \gamma, 0), \\ \widetilde{\mathbf{q}} &= \max(\breve{\mathbf{u}} - \gamma, 0). \end{aligned} \tag{2.16}$$

Therefore, if we obtain a dual optimal solution γ^* that maximizes the dual problem (2.15), then we can readily compute the closed form solution to (2.11) and thus to (2.10). That is, $\mathbf{w}^* = \mathbf{s}^{(0)} \circ \widetilde{\mathbf{w}}^*$, $W^*_{\cdot,j} = \mathbf{s}^{(j)} \circ \widetilde{W}^*_{\cdot,j}$ where $\mathbf{s}^{(0)} = \text{sign}(v_j)$, $\mathbf{s}^{(j)} = \text{sign}\left(U_{\cdot,j}\right)$, $j = 1, \ldots, d$, $\widetilde{\mathbf{w}}^*$ and columns of $\widetilde{W}^*$ are obtained via (2.16) at the optimal dual solution γ^*.

2.3.2 The Dual Optimal Solution

Next, we show how to efficiently compute the dual optimal solution γ^*. First, we sort $-\breve{v}$ and $\breve{u}_i, i = 1, \ldots, d$ in ascending order. Without loss of generality, we assume:

$$\breve{u}_1 \leq \ldots \leq \breve{u}_L \leq -\breve{v} \leq \breve{u}_{L+1} \leq \ldots \leq \breve{u}_d. \tag{2.17}$$

There are four possible cases about the locations of γ. We discuss how to identify the optimal dual solution γ^* in each of the four cases.

Case 1 :

When $\ldots \leq \breve{u}_G \leq \gamma \leq \breve{u}_{G+1} \leq \ldots \leq -\breve{v} \leq \ldots$, the objective in (2.15) at γ^* becomes

$$\begin{aligned} &\frac{1}{2}\sum_{i=1}^{G}(\breve{u}_i - \gamma)^2 + \frac{1}{2}(\breve{v} + \gamma)^2 + h(\gamma) \\ =\ &\frac{1}{2}\sum_{i=1}^{G}\breve{u}_i^2 + \sum_{i=G+1}^{d}\gamma\breve{u}_i - \frac{d-G}{2}\gamma^2 + \frac{1}{2}\breve{v}^2. \end{aligned} \tag{2.18}$$

Function (2.18) is a quadratic function with respect to γ and the unconstrained maximum is achieved at the axis of symmetry point $\frac{\sum_{i=G+1}^{d}\breve{u}_i}{d-G} \geq \breve{u}_{G+1}$. Since γ falls in the interval $[\breve{u}_G, \breve{u}_{G+1}]$, we set

$$\gamma = \breve{u}_{G+1}$$

to achieve the maximum objective value of (2.15). It can be further concluded that, in **Case 1**, among all the intervals on the left of $-\breve{v}$, the maximum objective value of (2.15) is achieved at the $\breve{u}_G$.

Case 2:

When $\ldots \leq \breve{u}_L \leq \gamma \leq -\breve{v} \leq \breve{u}_{L+1} \leq \ldots$, it turns out that the objective value in (2.15) at γ is similar to (2.18):

$$\frac{1}{2}\sum_{i=1}^{L}\breve{u}_i^2 + \sum_{i=L+1}^{d}\gamma\breve{u}_i - \frac{d-L}{2}\gamma^2 + \frac{1}{2}\breve{v}^2. \tag{2.19}$$

By a similar argument, we can set $\gamma = -\breve{v}$ to achieve the maximum. Combining the results of **Case 1** and **Case 2**, we conclude that, we may only consider γ in the range $[\max(-\breve{v}, 0), +\infty]$. Note that when $L = d$, that is $\breve{u}_d \leq \gamma \leq -\breve{v}$, (2.19) is a constant $\frac{1}{2}\sum_{i=1}^{d}\breve{u}_i^2 + \frac{1}{2}\breve{v}^2$, and thus γ can be any value in the interval $[\breve{u}_d, -\breve{v}]$.

Case 3:

When $\ldots \leq \breve{u}_L \leq -\breve{v} \leq \gamma \leq \breve{u}_{L+1} \leq \ldots$, the value of the objective function in (2.15)

at γ^* becomes

$$\begin{aligned} &\frac{1}{2}\sum_{i=1}^{L}(\breve{u}_i-\gamma)^2+h(\gamma) \\ =&\frac{1}{2}\sum_{i=1}^{L}\breve{u}_i^2+\gamma\left(\sum_{i=L+1}^{d}\breve{u}_i-\breve{v}\right)-\frac{d+1-L}{2}\gamma^2. \end{aligned} \tag{2.20}$$

Again, (2.20) is a quadratic function of γ and $\frac{\sum_{i=L+1}^{d}\breve{u}_i-\breve{v}}{d+1-L}\geq-\breve{v}$. If $\frac{\sum_{i=L+1}^{d}\breve{u}_i-\breve{v}}{d+1-L}\geq\breve{u}_{L+1}$, the maximum is achieved at

$$\gamma=\breve{u}_{L+1},$$

otherwise the maximum is achieved at

$$\gamma=\frac{\sum_{i=L+1}^{d}\breve{u}_i-\breve{v}}{d+1-L}.$$

Case 4:

When $\ldots\leq-\breve{v}\leq\ldots\leq\breve{u}_G\leq\gamma\leq\breve{u}_{G+1}\leq\ldots$, the objective value in (2.15) is similar to (2.20):

$$\frac{1}{2}\sum_{i=1}^{G}\breve{u}_i^2+\gamma\left(\sum_{i=G+1}^{d}\breve{u}_i-\breve{v}\right)-\frac{d+1-G}{2}\gamma^2. \tag{2.21}$$

If $\frac{\sum_{i=G+1}^{d}\breve{u}_i-\breve{v}}{d+1-G}\geq\breve{u}_{G+1}$, then the maximum is achieved at

$$\gamma=\breve{u}_{G+1};$$

If $\frac{\sum_{i=G+1}^{d}\breve{u}_i-\breve{v}}{d+1-G}\leq\breve{u}_G$, then the maximum is achieved at

$$\gamma=\breve{u}_G;$$

If $\breve{u}_G\leq\frac{\sum_{i=G+1}^{d}\breve{u}_i-\breve{v}}{d+1-G}\leq\breve{u}_{G+1}$, the maximum is achieved at

$$\gamma=\frac{\sum_{i=G+1}^{d}\breve{u}_i-\breve{v}}{d+1-G}.$$

Since we know exactly the value of γ for all the four cases, one naive way to find the optimal γ^* is to enumerate all the possible locations and pick the one that maximizes the objective function value in (2.15). However, evaluating the objectives for all possible locations from $\max(-\breve{v}, 0)$ to $\breve{u}_d$ leads to a quadratic time algorithm for solving (2.15). Interestingly, we show below that the time complexity of solving (2.15) can be reduced to $\mathcal{O}(d\log(d))$.

Let us first list some useful properties as follows:
Given the ordered sequence (2.17):

- **Property 1:**

  The maximum objective value of (2.15) in Case 3 is larger than the one in Cases 1 & 2;

- **Property 2:**

  In Case 4, for a pair of adjacent intervals $[\breve{u}_{G-1}, \breve{u}_G]$ and $[\breve{u}_G, \breve{u}_{G+1}]$, if $\frac{\sum_{i=G+1}^{d}\breve{u}_i-\breve{v}}{d+1-G} \geq \breve{u}_{G+1}$ for $[\breve{u}_G, \breve{u}_{G+1}]$, then $\frac{\sum_{i=G}^{d}\breve{u}_i-\breve{v}}{d+1-(G-1)} \geq \breve{u}_G$ for $[\breve{u}_{G-1}, \breve{u}_G]$;

- **Property 3:**

  In Case 4, if $\frac{\sum_{i=G+1}^{d}\breve{u}_i-\breve{v}}{d+1-G} \geq \breve{u}_{G+1}$ for $[\breve{u}_G, \breve{u}_{G+1}]$, the maximum objective value of (2.15) in $[\breve{u}_G, \breve{u}_{G+1}]$ is larger than or equal to the one in $[\breve{u}_{G-1}, \breve{u}_G]$.

- **Property 4:** In Case 4, for a pair of adjacent intervals $[\breve{u}_{G-1}, \breve{u}_G]$ and $[\breve{u}_G, \breve{u}_{G+1}]$, if we have $\frac{\sum_{i=G}^{d}\breve{u}_i-\breve{v}}{d+1-(G-1)} \leq \breve{u}_{G-1}$ for $[\breve{u}_{G-1}, \breve{u}_G]$, then $\frac{\sum_{i=G+1}^{d}\breve{u}_i-\breve{v}}{d+1-G} \leq \breve{u}_G$ for $[\breve{u}_G, \breve{u}_{G+1}]$.

- **Property 5:**

  In Case 4, if $\frac{\sum_{i=G}^{d}\breve{u}_i-\breve{v}}{d+1-(G-1)} \leq \breve{u}_{G-1}$ for $[\breve{u}_{G-1}, \breve{u}_G]$, the maximum objective value of (2.15) in $[\breve{u}_{G-1}, \breve{u}_G]$, is larger than or equal to the one in $[\breve{u}_G, \breve{u}_{G+1}]$.

- **Property 6:**

  In Case 4, if $\breve{u}_G \leq \frac{\sum_{i=G+1}^{d}\breve{u}_i-\breve{v}}{d+1-G} \leq \breve{u}_{G+1}$ for $[\breve{u}_G, \breve{u}_{G+1}]$, then $\frac{\sum_{i=G}^{d}\breve{u}_i-\breve{v}}{d+1-(G-1)} \geq \breve{u}_G$ for

$[\breve{u}_{G-1}, \breve{u}_G]$ and $\frac{\sum_{i=G+2}^{d} \breve{U}_i - \breve{v}}{d+1-(G+1)} \leq \breve{u}_{G+1}$ for $[\breve{u}_{G+1}, \breve{u}_{G+2}]$, and the maximum value of (2.15) in the interval $[\breve{u}_G, \breve{u}_{G+1}]$ is larger than or equal to the ones in its neighbor intervals.

Properties 2-6 also apply for adjacent intervals $[-\breve{v}, \breve{u}_{L+1}]$ and $[\breve{u}_{L+1}, \breve{u}_{L+2}]$ discussed in Case 3.

We omit the proof of Properties 1-6 since they are direct applications of 1-D quadratic optimization. Property 1 indicates that it is sufficient for the algorithm to start searching γ^* from Case 3. Properties 2 & 3 imply that, for some interval, if the axis of symmetry is on the right hand side of the interval, then one only needs to consider the intervals to the right. Similarly, Properties 4 & 5 indicate that, for some interval, if the axis of symmetry is on the left hand side of the interval, then one only needs to consider the intervals to the left. Property 6 combined with Properties 1-5 imply that, for certain interval, if it contains the axis of symmetry, then γ^* is the axis of symmetry point. Thus, we can draw the following conclusion:

(1) if $\max(\breve{u}_d, -\breve{v}) < 0$, then

$$\gamma^* = 0;$$

(2) if $-\breve{v} > \breve{u}_d$, then

$$\gamma^* = \max(-\breve{v},\ 0);$$

(3) if $\breve{u}_G \leq \frac{\sum_{i=G+1}^{d} \breve{u}_i - \breve{v}}{d+1-G} \leq \breve{u}_{G+1}$ for a certain interval $[\breve{u}_G, \breve{u}_{G+1}]$, then

$$\gamma^* = \max\left(\frac{\sum_{i=G+1}^{d} \breve{u}_i - \breve{v}}{d+1-G},\ 0\right).$$

At each move, the axis of symmetry $\frac{\sum_{i=G+1}^{d} \breve{u}_i - \breve{v}}{d+1-G}$ can be calculated by a constant operation based on the value from the last step, and the time complexity of searching γ^* reduces from quadratic to $\mathcal{O}(d\log(d))$ as the computation is dominated by the sorting operation. Once γ^* is determined, we can compute $\widetilde{w}^*$ and $\widetilde{W}^*$ by (2.16). Note

Algorithm 1: Computation of the Proximal Operator of Weak Hierarchical Lasso

Input: $\mathbf{v} \in \mathbb{R}^d$, $U \in \mathbb{R}^{d \times d}$, $t \in \mathbb{R}_+$, $\lambda \in \mathbb{R}_+$

Output: $\mathbf{w} \in \mathbb{R}^d$, $W \in \mathbb{R}^{d \times d}$

$\breve{\mathbf{v}} = \text{sign}(\mathbf{v}) \circ \mathbf{v} - \frac{\lambda}{t}\mathbf{1}$; $\breve{U} = \text{sign}(U) \circ U - \frac{\lambda}{2t}\mathbf{1}\mathbf{1}^T$;
for $j = 1$ to d **do**
  $c = -\breve{v}_j$;
  Sort $\breve{U}_{\cdot,j}$ to get a sequence $\mathcal{S}$ in ascending order where $\mathcal{S}_1 \leq \mathcal{S}_2 \leq \ldots \leq \mathcal{S}_d$;
  if $\mathcal{S}_d < 0$ and $c < 0$ **then**
    $\widetilde{w}_j = 0$; $\widetilde{W}_{\cdot,j} = \mathbf{0}$;
  else
    if $\mathcal{S}_d < c$ **then**
      $\gamma = \max(c, 0)$;
    else
      $k = d$;
      while 1 **do**
        $c = c + \mathcal{S}_k$; $k = k - 1$; $\gamma = c/(d + 1 - k)$;
        if $\gamma \geq 0$ **then**
          if $\gamma \geq \mathcal{S}_k$ **then**
            break;
          end if
        else
          $\gamma = 0$; break;
        end if
      end while
    end if
    $\widetilde{w}_j = \max(\breve{v}_j + \gamma, \mathbf{0})$; $\widetilde{W}_{\cdot,j} = \max\left(\breve{U}_{\cdot,j} - \gamma, \mathbf{0}\right)$;
  end if
end for
$\mathbf{w} = \text{sign}(\mathbf{v}) \circ \widetilde{\mathbf{w}}$; $W = \text{sign}(U) \circ \widetilde{W}$;

that, the subproblem of the proximal operator associated with the convex relaxation (Bien *et al.*, 2013) is solved by searching for the dual variable in a different way with time complexity $\mathcal{O}(d^2)$.

Algorithm 2: The Efficient Weak Hierarchical Lasso Algorithm (eWHL)

Input: $X \in \mathbb{R}^{n\times d}$, $Z \in \mathbb{R}^{n\times(d\cdot d)}$, $\lambda \in \mathbb{R}_+$, $\eta > 1$

Output: $\mathbf{w} \in \mathbb{R}^d$, $W \in \mathbb{R}^{d\times d}$

1: Initialize $k \leftarrow 0$ and starting points $\mathbf{w}^{(0)}$ and $W^{(0)}$;

2: **repeat**

3: Choose the step size $t^{(k)}$ by the BB Rule

4: **repeat**

5: $\mathbf{v}^{(k)} = \mathbf{w}^{(k)} - \nabla_{\mathbf{w}}\ell\left(\mathbf{w}^{(k)}, W^{(k)}\right)/t^{(k)}$;

$U^{(k)} = U^{(k)} - \nabla_{W}\ell\left(\mathbf{w}^{(k)}, W^{(k)}\right)/t^{(k)}$;

Solve $\left(\mathbf{w}^{(k+1)}, W^{(k+1)}\right)$ by Algorithm 1 with input $\left(\mathbf{v}^{(k)}, U^{(k)}, t^{(k)}, \lambda\right)$;

$t^{(k)} \leftarrow \eta t^{(k)}$;

6: **until** line search criterion is satisfied

7: $k \leftarrow k + 1$

8: **until** stop criterion is satisfied

In summary, we reformulate the proximal operator for the original weak hierarchical Lasso by factorizing the unknown coefficients. The reformulated proximal operator is shown to admit a closed form solution, which enables directly solving the weak hierarchical Lasso problem. Moreover, the subproblem of the proximal operator can be computed efficiently with a time complexity of $\mathcal{O}(d\log(d))$. The detailed algorithm for solving the proximal operator (2.10) is described in Algorithm 1. We give the details of eWHL algorithm in Algorithm 2. Following (Gong *et al.*, 2013a), we choose the step size $t^{(k)}$ by the Barzilai-Borwein (BB) Rule.

2.4 Hierarchical Testing

Recently, a permutation-based method named TMIcor, for testing interactions has been proposed for binary classification problems (Simon and Tibshirani, 2012). In contrast to model-based methods which select conditional interactions, TMIcor starts from considering interactions from the "backward model" and identifies marginal interactions instead (Simon and Tibshirani, 2012). In TMIcor, the null hypothesis for an interaction between feature i and feature j is that the correlations between them in the two classes are the same, and the test statistic is the size of the difference of Fisher transformed sample correlation. Significant interactions in TMIcor are those with test statistic greater than a certain threshold and the cutoff value is chosen to meet a predefined false discovery rate (FDR) which is estimated via permutation.

Based on TMIcor, the optimization based testing framework named Convex Hierarchical Testing (CHT) was proposed for testing pairwise interactions in binary classification problems (Bien *et al.*, 2015). By imposing hierarchical constraints, the test statistics of main effects and interactions obtained from CHT satisfy weak hierarchical structures. Specifically, given feature $\mathbf{x} \in \mathbb{R}^d$ and outcome $y \in \{+1, -1\}$, the conditional distribution is modelled as (Simon and Tibshirani, 2012; Bien *et al.*, 2015):

$$\mathbf{x}|y = \mathcal{C} \sim N_d\left(\mu^{(\mathcal{C})}, \Sigma^{(\mathcal{C})}\right),$$

where $\mu^{(\mathcal{C})} \in \mathbb{R}^d$ and $\Sigma^{(\mathcal{C})} \in \mathbb{R}^{d\times d}$ are the class specific mean vector and covariance matrix. The null hypotheses of main effects and interactions are:

$$H_{0,i} : \mu_i^{(+1)} = \mu_i^{(-1)} \quad \text{for } 1 \le i \le d \qquad \text{(main effects)}$$
$$H_{0,ij} : \rho_{i,j}^{(+1)} = \rho_{i,j}^{(-1)} \quad \text{for } 1 \le i < j \le d \quad \text{(interactions)}$$

where $\rho_{i,j}^{(\mathcal{C})} = \left(\Sigma_{i,i}^{(\mathcal{C})}\Sigma_{j,j}^{(\mathcal{C})}\right)^{-\frac{1}{2}} \Sigma_{i,j}^{(\mathcal{C})}$ is the class specific correlation.

For testing $H_{0,i}$, the t-statistic is

$$v_i = \frac{\bar{x}_i^{(+1)} - \bar{x}_i^{(-1)}}{\sqrt{s_i^{(+1)}/n^{(+1)} + s_i^{(-1)}/n^{(-1)}}},$$

where $n^{(\mathcal{C})}$ is the sample size of class $\mathcal{C}$,

$$\bar{x}_i^{(\mathcal{C})} = \frac{1}{n^{(\mathcal{C})}} \sum_{k:y_k=\mathcal{C}} X_{ki}$$

and

$$s_i^{(\mathcal{C})} = \frac{1}{n^{(\mathcal{C})} - 1} \sum_{k:y_k=\mathcal{C}} \left(X_{ki} - \bar{x}_i^{(\mathcal{C})} \right)^2$$

are the class specific sample mean and variance. For testing $H_{0,ij}$, the Fisher transformed statistic would be

$$U_{i,j} = \left(\frac{1}{n^{(+1)} - 3} + \frac{1}{n^{(-1)} - 3} \right) \left[\arctan\left(\hat{\rho}_{i,j}^{(+1)} \right) - \arctan\left(\hat{\rho}_{i,j}^{(-1)} \right) \right],$$

where

$$\hat{\rho}_{i,j}^{(\mathcal{C})} = \left(n^{(\mathcal{C})} - 1 \right)^{-1} \sum_{k:y_k=\mathcal{C}} \left(X_{ki}^{(\mathcal{C})} - \bar{x}_i^{(\mathcal{C})} \right) \left(X_{kj}^{(\mathcal{C})} - \bar{x}_j^{(\mathcal{C})} \right) / \left(s_i^{(\mathcal{C})} s_j^{(\mathcal{C})} \right)$$

is the class specific sample correlation.

Suppose the testing of interactions is modeled via the Lasso formulation, i.e.,

$$\min_{\mathbf{w},W} \frac{1}{2} \sum_{j=1}^{d} (w_j - v_j)^2 + \frac{1}{2} \sum_{i,j=1}^{d} (W_{i,j} - U_{i,j})^2 + \lambda \|\mathbf{w}\|_1 + \lambda \sum_{i,j=1}^{d} |W_{i,j}|, \tag{2.22}$$

and define $\lambda_{\max}$ as the largest λ resulting in nonzero coefficient of the corresponding main effect or interaction:

$$\lambda_{\max,i} := \sup\{\lambda \geq 0 : \hat{w}_i(\lambda) \neq 0\},$$

$$\lambda_{\max,ij} := \sup\{\lambda \geq 0 : \widehat{W}_{i,j}(\lambda) \neq 0, \quad \widehat{W}_{j,i}(\lambda) \neq 0\},$$

where $\hat{w}_i(\lambda)$ and $\widehat{W}_{i,j}(\lambda)$ are the solutions to (2.22) given λ. Then, $\lambda_{\max}$'s for main effects and interactions would be the sizes of t-statistics $|v_j|, 1 \leq j \leq d$ and Fisher

transformed statistics, i.e., $|U_{i,j}|, 1 \leq i, j \leq d$. Note that $\lambda_{\max}$'s for interactions in (2.22) are exactly the test statistics used in TMIcor. However, the drawback of formulation (2.22) lies in the failure of incorporating structural information between main effects and interactions. Inspired by the relation between $\lambda_{\max}$'s and the statistic sizes, Bien *et al.* (2015) proposed to introduce hierarchical structures to the testing of pairwise interactions for two-class problems. When the weak hierarchical constraints are imposed to problem (2.22), i.e.,

$$\begin{aligned} \min_{\mathbf{w},W} \ & \frac{1}{2}\sum_{j=1}^{d}(w_j - v_j)^2 + \frac{1}{2}\sum_{i,j=1}^{d}(W_{i,j} - U_{i,j})^2 + \lambda\|\mathbf{w}\|_1 + \lambda\sum_{i,j=1}^{d}|W_{i,j}| \\ \text{s.t.} & \sum_{i=1}^{d}|W_{i,j}| \leq |w_j| \text{ for } j = 1, \ldots, d. \end{aligned} \tag{2.23}$$

and the test statistics for main effects and interactions are defined as the corresponding $\lambda_{\max}$'s, then a weak hierarchical structure exists among the test statistics, i.e.,

$$\lambda_{\max,ij} \leq \max\left(\lambda_{\max,i}, \lambda_{\max,j}\right).$$

Instead of directly solving problem (2.23), Bien *et al.* (2015) relaxed the non-convex constraints and propose its convex relaxation:

$$\begin{aligned} \min_{\mathbf{w}^+,\mathbf{w}^-,W} \ & \frac{1}{2}\sum_{j=1}^{d}\left(w_j^+ - w_j^- - v_j\right)^2 + \frac{1}{2}\sum_{i,j=1}^{d}(W_{i,j} - U_{i,j})^2 \\ & + \lambda\sum_{j=1}^{d}\left(w_j^+ + w_j^-\right) + \lambda\sum_{i,j=1}^{d}|W_{i,j}| \\ \text{s.t.} \ & \sum_{i=1}^{d}|W_{i,j}| \leq w_j^+ + w_j^-, \\ & w_j^+ \geq 0, \\ & w_j^- \geq 0 \qquad \text{for} \quad j = 1, \ldots, d. \end{aligned} \tag{2.24}$$

Note that the formulation of CHT is essentially equivalent to that of the proximal operator associated with the convex weak hierarchical Lasso. Bien *et al.* (2015) derived the closed form expression of the test statistics for CHT.

Instead of using the convex relaxation (2.24), we propose to adopt the non-convex formulation (2.23) directly and use the corresponding $\lambda_{\max}$'s for the significance testing of pairwise interactions. As problem (2.23) is essentially equivalent to the proximal operator problem (2.10) associated with weak Hierarchical Lasso (2.3), we can directly solve it as detailed in Section 3. We name the testing framework using formulation (2.23) as the **n**on-**C**onvex **H**ierarchical **T**esting (nCHT).

For solving (2.23), we have shown that

$$\begin{aligned} \text{sign}(v_i) &= \text{sign}(w_i), \\ \text{sign}(W_{i,j}) &= \text{sign}(U_{i,j}), \end{aligned} \tag{2.25}$$

and thus we only need to solve

$$\begin{aligned} \min_{\mathbf{w}\succeq 0, W\succeq 0} \quad & \frac{1}{2}\sum_{j=1}^{d}\left(w_j - |v_j|\right)^2 + \frac{1}{2}\sum_{i,j=1}^{d}\left(W_{i,j} - |U_{i,j}|\right)^2 + \lambda \mathbf{1}^T\mathbf{w} + \lambda\sum_{i,j=1}^{d} W_{i,j} \\ \text{s.t.} \quad & \sum_{i=1}^{d} W_{i,j} \le w_j \qquad \text{for } j = 1,\ldots,d. \end{aligned} \quad . \tag{2.26}$$

Since the solutions to (2.26) are the magnitudes of those to (2.23), the $\lambda_{\max}$'s are equivalent for the two formulations. Therefore, it is sufficient to only study the solution path of problem (2.26).

Again, a brute force search for $\lambda_{\max}$'s is computationally intensive. We will show that the $\lambda_{\max}$'s for both main effects and interactions admit closed-form solutions which can greatly speed up the computation of test statistics. Similar to the analysis of problem (2.10), problem (2.26) can be decoupled to d subproblems:

$$\begin{aligned} \min_{w\ge 0, \mathbf{q}\succeq 0} \quad & \frac{1}{2}\left(w - |v|\right)^2 + \frac{1}{2}\sum_{i=1}^{d}\left(q_i - |u_i|\right)^2 + \lambda w + \lambda\sum_{i=1}^{d} q_i \\ \text{s.t.} \quad & \sum_{i=1}^{d} q_i \le w \end{aligned} \quad , \tag{2.27}$$

where we simplify the notations by omitting the subscript j on w and v and write $\mathbf{q}, \mathbf{u} \in \mathbb{R}^d$ for $W_{\cdot,j}$ and $U_{\cdot,j}$ respectively. We first state the basic observations of $\lambda_{\max}$

in different cases in Lemma 2.4.1 and show the closed-form solutions to $\lambda_{\max}$'s in Theorem 2.4.2 .

Lemma 2.4.1. *The $\lambda_{\max}$'s for both w and $\mathbf{q}$ have the following forms:*

1. *If $\|\mathbf{u}\|_\infty \leq |v|$, then*

$$\lambda_{\max}^{w} = |v|$$

$$\lambda_{\max}^{q_i} = \begin{cases} |u_i| & \text{if } |u_i| \geq \bar{\lambda} \\ \frac{1}{2}\left(|v| + |u_i| - \sum_{j=i}^{d}(|u_j| - |u_i|)\right)_+ & \text{if } |u_i| < \bar{\lambda} \end{cases}$$

2. *If $\|\mathbf{u}\|_\infty > |v|$, then*

$$\lambda_{\max}^{w} = \frac{1}{2}(|v| + \|\mathbf{u}\|_\infty)$$

$$\lambda_{\max}^{W_i} = \frac{1}{2}\left(|v| + |u_i| - \sum_{j=i}^{d}(|u_j| - |u_i|)\right)_+$$

where $(\cdot)_+$ is the thresholding operator, i.e. $(\alpha)_+ = \alpha$ if $\alpha \geq 0$ and $(\alpha)_+ = 0$ if $\alpha < 0$, and $\bar{\lambda}$ is defined as:

$$\bar{\lambda} := \max\left\{0 \leq \lambda \leq \|\mathbf{u}\|_\infty : \sum_{i=1}^{d}(|u_i| - \lambda)_+ + \lambda \geq |v|\right\}.$$

Proof. In the proof, we denote $\breve{v} = |v| - \lambda$ and $\breve{u}_i = |u_i| - \lambda$ and assume the sizes of the elements in $\mathbf{u}$ are in an ascending order, i.e., $|u_1| \leq |u_2| \leq \ldots \leq |u_{d-1}| \leq |u_d|$. We next discuss the solutions to problem (2.27) with varying penalty parameter value λ.

CASE I: $\|\mathbf{u}\|_\infty \leq |v|$.

(i) $\|\mathbf{u}\|_\infty \leq |v| \leq \lambda$:

In this case, we have $-\breve{v} \geq 0 > \breve{u}_d$. It is straightforward to verify that

$$w = 0 \text{ and } q_i = 0, i = 1, \ldots, d. \tag{2.28}$$

(ii) $\|\mathbf{u}\|_\infty < \lambda < |v|$:

$\|\mathbf{u}\|_\infty < \lambda < |v|$ if and only if $\breve{u}_d$ and $-\breve{v}$ are both negative, which results in $\gamma^* = 0$ based on previous analysis. Therefore, according to (2.16), we have

$$w = |v| - \lambda > 0 \text{ and } q_i = (|u_i| - \lambda)_+ = 0. \tag{2.29}$$

So far, we have considered the case where $\lambda > \|\mathbf{u}\|_\infty$. Before discussing the case where $\lambda \leq \|\mathbf{u}\|_\infty \leq |v|$, we list the following facts that are straightforward to verify:

$$\begin{array}{ll} 0 \leq \bar{\lambda} \leq |u_{d-1}| & \text{if } \|u\|_1 \geq |v| \\ \bar{\lambda} \in \emptyset & \text{if } \|u\|_1 < |v| \end{array}.$$

(iii) $0 \leq \lambda \leq \|\mathbf{u}\|_\infty \leq |v|$:

(a) $0 \leq \lambda \leq \|\mathbf{u}\|_\infty \leq |v| \leq \|\mathbf{u}\|_1$:

When $\lambda > \bar{\lambda}$, $\frac{\sum_i^d \breve{u}_i - \breve{v}}{d+2-i} < 0$ and $\gamma^* = 0$. It can be verified that

$$w > 0 \text{ and } q_i = (|u_i| - \lambda)_+. \tag{2.30}$$

Thus, one must have $\lambda = |u_i| > \bar{\lambda}$ to make $q_i = 0$.

When $0 \leq \lambda \leq \bar{\lambda}$, $\gamma^* = \frac{\sum_i^d \breve{u}_i - \breve{v}}{d+2-i} \geq 0$. In this case, we have

$$w > 0 \text{ and } q_i = (|u_i| - \lambda - \gamma^*)_+. \tag{2.31}$$

If $\gamma^* > \breve{u}_i$ for some $\breve{u}_i$, then $q_i = 0$ always holds. If $\gamma^* \leq \breve{u}_i$ for some $\breve{u}_i$, then

$$\lambda \leq \frac{1}{2}\left(|v| + |u_i| - \sum_{j=i}^{d}(|u_j| - |u_i|)\right).$$

As λ is non-negative, we have

$$\frac{1}{2}\left(|v| + |u_i| - \sum_{j=i}^{d}(|u_j| - |u_i|)\right) \geq 0.$$

Also, since γ^* is non-negative, we have $\breve{u}_i \geq 0$, i.e., $|u_i| \geq \lambda$. When

$$|u_i| \geq \frac{1}{2}\left(|v| + |u_i| - \sum_{j=i}^{d}(|u_j| - |u_i|)\right),$$

i.e., $|u_i| \leq \bar{\lambda}$, we obtain that λ needs to be $\frac{1}{2}\left(|v| + |u_i| - \sum_{j=i}^{d}(|u_j| - |u_i|)\right)$ such that $q_i = 0$. If $|u_i| < \frac{1}{2}\left(|v| + |u_i| - \sum_{j=i}^{d}(|u_j| - |u_i|)\right)$, i.e., $|u_i| > \bar{\lambda}$, then $q_i > 0$ always holds.

(b) $0 \leq \lambda \leq \|\mathbf{u}\|_\infty \leq \|\mathbf{u}\|_1 \leq |v|$

In this case, we have $\frac{\sum_i^d \breve{u}_i - \breve{v}}{d+2-i} < 0$. It follows that

$$w > 0 \text{ and } q_i = (|u_i| - \lambda)_+, \tag{2.32}$$

and λ needs to be $|u_i|$ such that $q_i = 0$.

Based on (2.28), (2.29), (2.30), (2.31), (2.32), when $\|\mathbf{u}\|_\infty \leq |v|$ we obtain that

$$\lambda_{\max}^{w} = |v|$$

$$\lambda_{\max}^{q_i} = \begin{cases} |u_i| & \text{if } |u_i| \geq \bar{\lambda} \\ \frac{1}{2}\left(|v| + |u_i| - \sum_{j=i}^{d}(|u_j| - |u_i|)\right)_+ & \text{if } |u_i| < \bar{\lambda} \end{cases}$$

CASE II: $\|\mathbf{u}\|_\infty > |v|$.

(i) $|v| < \frac{1}{2}(|v| + \|\mathbf{u}\|_\infty) \leq \lambda \leq \|\mathbf{u}\|_\infty$:

When $\lambda \geq \frac{1}{2}(|v| + \|\mathbf{u}\|_\infty)$, we have $-\breve{v} \geq \breve{u}_d$, and

$$w = 0 \text{ and } q_i = 0. \tag{2.33}$$

(i) $|v| \leq \lambda < \frac{1}{2}(|v| + \|\mathbf{u}\|_\infty) < \|\mathbf{u}\|_\infty$:

In this case, it can be verified that $w \neq 0$ always holds. We can make analogous conclusions for the solutions of q_i's based on similar analysis for CASE I with $\lambda \leq \|\mathbf{u}\|_\infty$. Note that, with $\|\mathbf{u}\|_\infty > |v|$, $\bar{\lambda} = \|\mathbf{u}\|_\infty$ and therefore $\frac{1}{2}(|v| + \|\mathbf{u}\|_\infty) \leq \bar{\lambda}$ always holds.

Thus, when $\|\mathbf{u}\|_\infty > |v|$, we obtain that

$$\lambda_{\max}^{w} = \frac{1}{2}(|v| + \|\mathbf{u}\|_\infty)$$

$$\lambda_{\max}^{q_i} = \frac{1}{2}\left(|v| + |u_i| - \sum_{j=i}^{d}(|u_j| - |u_i|)\right)_+ .$$

We thus complete the proof. □

We can directly obtain the following theorem from Lemma 2.4.1.

Theorem 2.4.2. *The $\lambda_{\max}$'s for coefficients w and $\mathbf{q}$ have the following closed-form solution:*

$$\lambda_{\max}^{w} = \max\left(|v|,\ \frac{|v| + \|\mathbf{u}\|_\infty}{2}\right)$$
$$\lambda_{\max}^{q_i} = \min\left(|u_i|,\ \frac{1}{2}\left(|v| + |u_i| - \sum_{j=i}^{d}\left(|u_j| - |u_i|\right)\right)_+\right).$$

For the testing of pairwise interactions for two-class problems, the hypotheses are rejected if their test statistics (i.e., $\lambda_{\max}$'s for $\mathbf{w}$ and W) are greater than the threshold at which the FDR is estimated to achieve a satisfied criterion. FDR is estimated by a scheme of permutation (Bien *et al.*, 2015; Simon and Tibshirani, 2012; Tusher *et al.*, 2001). At each permutation, we randomly shuffle the class labels $\mathbf{y}$ and re-compute v_i's and $U_{i,j}$'s. The permutation is made $\mathcal{B}$ times and the test statistics ${\lambda_{\max,ij}}^{(b)}$'s at the b-th permutation is computed. Then, the FDR at λ is estimated as

$$\widehat{\mathrm{FDR}}(\lambda) = \min\left\{\frac{\frac{1}{\mathcal{B}}\sum_{i,j,b}\mathcal{I}_{{\lambda_{\max,ij}}^{(b)}>\lambda}}{\sum_{i,j}\mathcal{I}_{\lambda_{\max,ij}>\lambda}}, 1\right\}, \tag{2.34}$$

where $\mathcal{I}$ is the indicator function of which function value is 1 if the corresponding condition is satisfied and 0 otherwise. In practice, FDRs are estimated at multiple λ's and the threshold is chosen as the one achieving a predefined FDR value (or below).

2.5 Experimental Results

In this section, we evaluate the efficiency and effectiveness of the proposed algorithm on both synthetic and real data sets. In our first experiment, we compare the efficiency of our proposed algorithm and the convex relaxation of weak hierarchical Lasso (Bien *et al.*, 2013) on synthetic data sets where the weak hierarchical structure holds between main effects and interaction effects. In our second experiment, we

compare the classification performance of the weak hierarchical Lasso with other classifiers and sparse learning techniques on the data collected from Alzheimer's Disease Neuroimaging Initiative (ADNI) [1] .

2.5.1 *Efficiency and Effectiveness Comparison on Synthetic Data Sets*

In this experiment, we compare the efficiency of the proposed eWHL algorithm with the convex relaxation on synthetic data sets. Our algorithm is built upon the GIST framework which is available online (Gong *et al.*, 2013b). The source code of the convex relaxed weak hierarchical Lasso (cvxWHL) was available in the R package "*hierNet*" (Bien and Tibshirani, 2013) where the optimization procedure was implemented by C. Since the proposed algorithm in this article directly solves the non-convex weak hierarchical Lasso (2.3), and the eventual goal of the convex relaxed weak hierarchical Lasso is also to find a good "relaxed" solution to the original problem, we compare the two algorithms in terms of the objective function in (2.3). In the experiment, entries of $X \in \mathbb{R}^{n\times d}$ are i.i.d generated from the standard normal distribution, *i.e.,* $X_{i,j} \sim N(0,1)$. The matrix of interactions, Z, is then generated via the normalized X where $Z = \left[Z^{(1)}, Z^{(2)}, \ldots, Z^{(d)}\right]$, $Z^{(i)} \in \mathbb{R}^{n\times d}$, $Z^{(i)}_{\cdot,j} = X_{\cdot,i} \circ X_{\cdot,j}$. The ground truths $w \in \mathbb{R}^{d\times 1}$ and $W \in \mathbb{R}^{d\times d}$ are generated based on the weak hierarchical structure $\|W_{\cdot,j}\|_1 \leq |w_j|, j = 1, \ldots, d$. In addition, we vary the ratio of coefficient sparsity, *i.e.,* the portion of zero entries in $\mathbf{w}$ and W, from 30% to 85%. Then, the outcome vector $\mathbf{y}$ is constructed as

$$\mathbf{y} = X\mathbf{w} + \frac{1}{2} Z \cdot \mathrm{vec}(W) + \epsilon$$

where X and Z are normalized to zero mean and unit standard deviation and $\epsilon \sim N(\mathbf{0}, 0.01 \cdot \mathbf{I})$. We use sample size $n = 100$ and 200 and we choose the number of main

[1] http://www.adni-info.org/

Table 2.1: Comparison of execution time (second) of the proposed algorithm for the non-convex weak hierarchical Lasso (eWHL) and the one for the convex relaxed formulation (cvxWHL) on synthetic data. The penalty parameters used in the experiment are from $\{1, 3, 5, 10, 20\}$. The data is generated under the weak hierarchical constraints where the portion of sparse coefficients is controlled to 85%. Two sample sizes, $n = 100$ and $n = 200$, are used and we vary the number of individual features from $\{200, 300, 400, 500, 600\}$ corresponding to $\{20100, 45150, 80200, 125250, 180300\}$ interactions (including the self product terms).

		n = 100					n = 200				
		85% Ground Truth Sparsity									
d	Methods	1	3	5	10	20	1	3	5	10	20
200	cvxWHL	196.5536	54.8801	43.3018	27.2806	15.7909	116.8207	24.6601	17.8850	9.1765	4.7783
	eWHL	15.9318	10.7613	7.2212	5.6287	2.6236	16.4134	9.5164	8.3827	5.4922	3.9255
	speedup	**12.3**	**5.1**	**6.0**	**4.8**	**6.0**	**7.1**	**2.6**	**2.1**	**1.7**	**1.2**
300	cvxWHL	336.7086	213.7712	186.7997	109.7893	54.9521	319.6003	147.5044	112.5928	59.0820	36.2484
	eWHL	35.6846	23.3044	17.9931	11.5569	10.8269	38.5045	20.0161	16.4566	10.3588	6.9153
	speedup	**9.4**	**9.2**	**10.4**	**9.5**	**5.1**	**8.3**	**7.4**	**6.8**	**5.7**	**5.2**
400	cvxWHL	547.0450	280.6981	207.8486	170.4894	85.1425	921.7651	376.4949	256.8054	144.3066	81.4817
	eWHL	52.8138	35.0482	29.5107	18.1944	13.8530	80.4882	54.1618	39.1673	26.6412	14.7667
	speedup	**10.4**	**8.0**	**7.0**	**9.4**	**6.1**	**11.5**	**7.0**	**6.6**	**5.4**	**5.5**
500	cvxWHL	1018.9779	757.2096	524.9644	333.0070	204.2017	1405.9651	1142.2343	964.0598	286.2120	165.2558
	eWHL	88.0526	66.0113	59.7805	42.2917	18.6453	127.0921	89.1293	70.0550	42.0014	29.0936
	speedup	**11.6**	**11.5**	**8.8**	**7.9**	**11.0**	**11.1**	**12.8**	**13.8**	**6.8**	**5.7**
600	cvxWHL	2543.5021	1594.9729	1517.9605	887.8254	462.2604	2826.0083	1558.1431	1332.3515	873.6990	261.6806
	eWHL	161.7944	100.3758	82.7961	71.1211	40.9529	197.5593	132.2163	107.5831	76.1834	45.0594
	speedup	**15.7**	**15.9**	**18.3**	**12.5**	**11.3**	**14.3**	**11.8**	**12.4**	**11.5**	**5.8**

Table 2.2: Comparison of execution time (second) of the proposed algorithm for the non-convex weak hierarchical Lasso (eWHL) and the one for the convex relaxed formulation (cvxWHL) on synthetic data. The penalty parameters used in the experiment are from $\{1, 3, 5, 10, 20\}$. The data is generated under the weak hierarchical constraints where the portion of sparse coefficients is controlled to 60%. Two sample sizes, $n = 100$ and $n = 200$, are used and we vary the number of individual features from $\{200, 300, 400, 500, 600\}$ corresponding to $\{20100, 45150, 80200, 125250, 180300\}$ interactions (including the self product terms).

		n = 100					n = 200				
		60% Ground Truth Sparsity									
200	cvxWHL	106.6262	40.3105	29.1357	20.8624	10.3064	113.3342	44.1169	27.2844	18.7616	11.9756
	eWHL	15.1405	9.6837	6.9516	5.4949	3.3569	18.3514	10.5571	8.1777	5.1257	4.1127
	speedup	**7.0**	**4.2**	**4.2**	**3.8**	**3.1**	**6.2**	**4.2**	**3.3**	**3.7**	**2.9**
300	cvxWHL	187.7983	131.7578	106.2882	61.3653	38.3189	290.0877	155.0435	131.7942	85.8886	44.4029
	eWHL	33.3861	22.2763	16.3251	12.3395	9.2993	47.8122	26.1554	21.9835	13.7322	10.5702
	speedup	**5.6**	**5.9**	**6.5**	**5.0**	**4.1**	**6.1**	**5.9**	**6.0**	**6.3**	**4.2**
400	cvxWHL	418.9647	276.2089	169.4631	131.9086	84.4169	686.8900	297.7161	226.6632	166.2235	85.4570
	eWHL	66.8376	43.0676	35.7516	20.5413	11.9166	69.1413	41.3634	37.3495	25.9975	16.0270
	speedup	**6.3**	**6.4**	**4.7**	**6.4**	**7.1**	**9.9**	**7.2**	**6.1**	**6.4**	**5.3**
500	cvxWHL	1501.3934	801.0146	548.8402	362.7110	206.0816	1333.8803	861.6311	729.1297	310.6121	202.0412
	eWHL	112.5519	80.5276	60.2488	38.6862	25.5497	114.5243	73.6990	63.4899	47.9597	31.5058
	speedup	**13.3**	**9.9**	**9.1**	**9.4**	**8.1**	**11.6**	**11.7**	**11.5**	**6.5**	**6.4**
600	cvxWHL	1976.0945	1733.8494	1814.1974	815.4298	323.4841	1622.9061	1205.8489	987.4595	1063.2823	333.8406
	eWHL	159.8307	112.8232	80.3703	50.8628	34.4373	175.9730	140.7086	96.3447	73.7633	52.2213
	speedup	**12.4**	**15.4**	**22.6**	**16.0**	**9.4**	**9.2**	**8.6**	**10.2**	**14.4**	**6.4**

Table 2.3: Comparison of execution time (second) of the proposed algorithm for the non-convex weak hierarchical Lasso (eWHL) and the one for the convex relaxed formulation (cvxWHL) on synthetic data. The penalty parameters used in the experiment are from $\{1, 3, 5, 10, 20\}$. The data is generated under the weak hierarchical constraints where the portion of sparse coefficients is controlled to 30%. Two sample sizes, $n = 100$ and $n = 200$, are used and we vary the number of individual features from $\{200, 300, 400, 500, 600\}$ corresponding to $\{20100, 45150, 80200, 125250, 180300\}$ interactions (including the self product terms).

		n = 100					n = 200				
		30% Ground Truth Sparsity									
	cvxWHL	139.4226	116.3866	85.1606	50.2425	23.8680	223.2468	165.9219	52.1613	40.8929	25.3084
200	eWHL	18.5023	13.6312	8.7261	7.1346	4.4546	20.8344	15.0214	9.7081	6.8616	4.0905
	speedup	**7.5**	**8.5**	**9.8**	**7.0**	**5.4**	**10.7**	**11.0**	**5.4**	**6.0**	**6.2**
	cvxWHL	275.4393	162.5627	139.4590	79.0609	41.2985	575.3758	223.1688	205.4368	142.9171	97.5106
300	eWHL	41.4815	27.6094	23.3748	14.2635	7.9047	52.3714	33.1059	25.5594	15.9962	10.6426
	speedup	**6.6**	**5.9**	**6.0**	**5.5**	**5.2**	**11.0**	**6.7**	**8.0**	**8.9**	**9.2**
	cvxWHL	916.5276	510.0533	342.0358	208.9260	104.5098	1688.7030	814.6646	560.1970	332.3118	204.4852
400	eWHL	75.6108	47.3789	38.8362	23.9130	17.6649	92.1627	60.4650	45.9561	34.6420	23.9751
	speedup	**12.1**	**10.8**	**8.8**	**8.7**	**5.9**	**18.3**	**13.5**	**12.2**	**9.6**	**8.5**
	cvxWHL	1460.8334	900.1424	767.7501	576.6498	242.9080	2003.7611	2040.7488	1632.3245	584.2366	313.8258
500	eWHL	114.9244	71.4278	58.4604	37.5124	25.9513	154.1934	102.7105	84.2729	63.9484	35.2531
	speedup	**12.7**	**12.6**	**13.1**	**15.4**	**9.4**	**13.0**	**19.9**	**19.4**	**9.1**	**8.9**
	cvxWHL	2799.5549	2842.9022	2076.6074	1148.0632	460.4660	4067.0519	2795.7589	2128.9981	1946.2750	1140.4244
600	eWHL	186.1483	119.8450	85.9816	65.2515	41.2607	165.9264	179.4403	146.7908	102.6978	71.8432
	speedup	**15.0**	**23.7**	**24.2**	**17.6**	**11.2**	**24.5**	**15.6**	**14.5**	**19.0**	**15.9**

effects d from $\{100,\ 200,\ 300,\ 400,\ 500,\ 600\}$. The parameter of the l_1 penalty, λ, is chosen from $\{1,\ 3,\ 5,\ 10,\ 20\}$. All algorithms are executed on a 64-bit machine with Intel(R) Core(TM) quad-core processor (i7-3770 CPU @ 3.40 GHz) and 16.0 GB memory. We terminate the algorithm when the maximum relative difference of the coefficients between two consecutive iterations is less than $1e-5$. We run 20 trials for each setting and report the average execution time. The detailed results are shown in Tables 2.1, 2.2 and 2.3.

From Tables 2.1, 2.2 and 2.3, we observe that eWHL is significantly faster than cvxWHL. Our algorithm is up to 25 times faster than the competing algorithm. As the dimension increases, the running time of cvxWHL increases much faster than our proposed algorithm. Specifically, when the number of individual features increases to 400 (corresponds to 80200 interactions), cvxWHL may take more than one thousand seconds, while the proposed eWHL is reasonably fast even when the number of total variables is around two hundred thousands.

To make further comparisons of efficiency, we randomly generate three synthetic datasets where the weak hierarchical structure between main effects and interactions holds. The three datasets are of the same sample size $n = 100$ and the number of individual features is $d = 300$. The ratios of zero entries in the ground truth are 85%, 60% and 30% respectively. The regularization parameters are chosen from $\{0.5,\ 1,\ 2,\ 4,\ 6,\ 8,\ 16,\ 32,\ 64\}$. On each dataset, we first run cvxWHL, and then the objective value of (2.3) in the final step is recorded. Then, we run the proposed eWHL and terminate the algorithm when the objective value of (2.3) is less than the one obtained by cvxWHL. The running time and the number of iterations needed to achieve the same objective value of both algorithms are reported in Figure 2.1. We observe from that the proposed eWHL is much faster than cvxWHL.

Moreover, we also conduct an experiment to compare the recovery performance

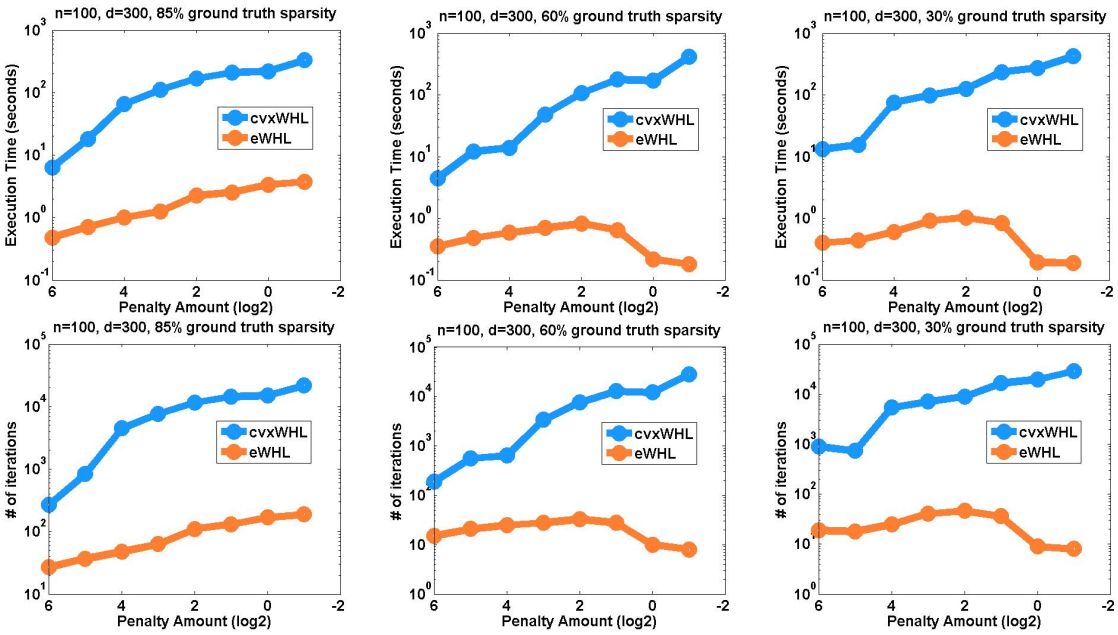

Figure 2.1: Comparison of the running time and the number of iterations by the two algorithms. Three synthetic data sets are generated where the portions of zeros in the ground truth are 85%, 60%, 30% respectively. The plots in the same column correspond to the same data set. The plots in the first row present the running time and those in the second row show the number of iterations.

of eWHL and cvxWHL. We generate synthetic data sets with sample size $n = 100$ and the number of individual features is $d = 50$ (1225 cross interactions). The number of non-zero main effects varies from $\{3, 4, 5, 6, 7\}$ and the number of non-zero interaction effects is from $\{2, 4, 5, 8, 10\}$, respectively. For each setting, ten synthetic data sets are generated with noise $\epsilon \sim N(\mathbf{0}, 0.01 \cdot \mathbf{I})$. We run both eWHL and cvxWHL with parameter selected via 5-fold cross-validation. Then we compute the sensitivity and specificity of recovery (where non-zero entries are positive and zero entries are negative). The means of sensitivity and specificity are plotted in Figure 2.2. We can observe that both algorithms achieve high recovery rate while directly solving the original weak hierarchical Lasso leads to slightly better performance in recovering the

non-zero effects.

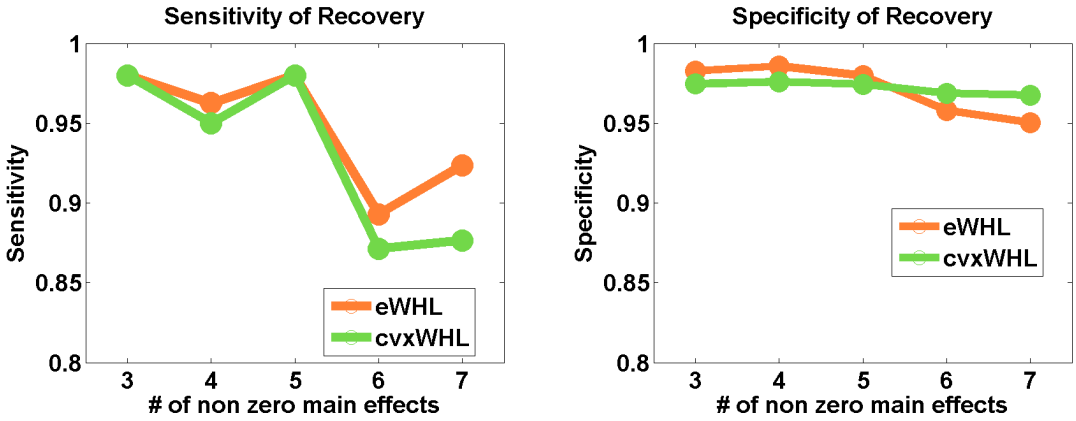

Figure 2.2: Comparison of eWHL and cvxWHL in terms of recovery on synthetic data sets.

2.5.2 Classification Comparisons on ADNI Data

In this experiment, we compare the weak hierarchical Lasso with its convex relaxation as well as other classifiers on the Alzheimer's Disease Neuroimaging Initiative (ADNI) data set.

In Alzheimer's Disease (AD) research, Mild Cognitive Impairment (MCI) is an intermediate state between normal elderly people and AD patients Petersen (2003). The MCI patients are considered to be at high risk of progression to AD. Many recent work focus on how to accurately predict the MCI-AD conversion and identifying significant bio-markers for the prediction (Davatzikos *et al.*, 2011; Devanand *et al.*, 2007; Fennema-Notestine *et al.*, 2009; Li *et al.*, 2014a; Llano *et al.*, 2011; Ye *et al.*, 2012; Zhou *et al.*, 2013; Gong *et al.*, 2012).

In this experiment, we compare the classification performance of the proposed eWHL with the convex relaxation and other classifiers on the task of discriminating the MCI subjects who convert to dementia (*i.e.,* MCI converter) within a three-

Table 2.4: The statistics of the ADNI data set used in our experiment. The MCI converters (MCI-cvt) are characterized as positive samples and the MCI non-converters (MCI non-cvt) are used as negative samples.

	Total	(+) MCI-cvt	(-) MCI non-cvt
# of samples	133	71	62
# of main effects	36		
# of interactions	630		

year period from the MCI subjects who remain at MCI (*i.e.,* MCI non-converter). The features used in the experiment (provided by our clinical collaborators) involve demographic information such as age, gender, years of education, clinical information such as scores of mini mental state examination (MMSE), Auditory Verbal Learning Test (A.V.L.T.), and the bio-markers including status of Apolipoprotein E, volume of hippocampus, thickness of Mid Temporal Gray Matter. There are 133 samples in total and the number of individual features is 36 (corresponds to 630 two way interactions). The interactions are generated by the normalized individual features and are normalized before entering the model. Since this is a classification task with binary labels, we replace the least square loss with logistic loss in the weak hierarchical Lasso. Besides the non-convex and convex weak hierarchical Lasso, we apply random forest (RF), Support Vector Machine (SVM) and sparse logistic regression on main effects, and on both main effects and interactions, respectively. We report the means and standard deviations of accuracy, sensitivity and specificity obtained from 10-fold cross-validation. The penalty parameters are tuned via 5-fold cross-validation in the training procedure. The sample statistics are shown in Table 2.4 and the classification performance is reported in Table 2.5.

Table 2.5: The performance of MCI converter vs. MCI non-converter classification achieved by random forest (RF), Support Vector Machine (SVM), Sparse Logistic Regression (spsLog), the convex relaxed weak hierarchical Lasso (cvxWHL) and the proposed algorithm (eWHL). Classifiers are performed on main effects only (top) and on both the main effects and interactions (bottom). The average and standard deviation of accuracy, sensitivity and specificity obtained from 10-fold cross-validation are reported.

Main Effects Only					
	RF	SVM	spsLog	cvxWHL	eWHL
Accuracy (%)	74.23 ± 8.67	75.22 ± 8.72	74.34 ± 9.56	NA	NA
Sensitivity (%)	78.75 ± 14.00	80.18 ± 13.89	80.18 ± 13.88	NA	NA
Specificity (%)	69.29 ± 11.63	69.76 ± 12.80	69.52 ± 13.74	NA	NA
Main Effects + Interactions					
	RF	SVM	spsLog	cvxWHL	eWHL
Accuracy (%)	71.26 ± 10.22	59.45 ± 14.43	73.57 ± 10.30	75.22 ± 11.02	77.42 ± 8.50
Sensitivity (%)	83.04 ± 13.18	59.29 ± 17.83	74.29 ± 16.22	75.71 ± 19.11	77.14 ± 12.05
Specificity (%)	58.10 ± 23.23	60.00 ± 15.42	72.86 ± 12.46	74.52 ± 16.84	77.62 ± 15.02

From Table 2.5, we can observe that, if we only use individual features for classification, then all the classifiers are biased towards the positive class, *i.e.,* MCI converter. When interactions are included, we observe that the performances of random forest and SVM become worse. One possible reason is that the large number of variables brought by the interactions weakens their discriminative power. This is not the case for sparse logistic regression, which demonstrates the importance of feature selection. We can observe from the table that the convex relaxed weak hierarchical Lasso and the non-convex weak hierarchical Lasso achieve much better classification performance than the competitors. The improvement of the classification performance demonstrates the effectiveness of imposing hierarchical structures in interaction models. In addition, the superior classification performance (around 77% accuracy, sensitivity and specificity) of the proposed eWHL demonstrates that directly solving the non-convex weak hierarchical Lasso leads to solutions of higher quality than the convex relaxation.

2.5.3 Simulation Studies for Hierarchical Testing

In this experiment, we compare the testing power of nCHT, CHT and the method using only Lasso type formulation on synthetic data where the weak hierarchy holds. Specifically, we generate 300 data points in total from Gaussian distribution where half of them belong to the positive class (+1) and the remaining half belongs to the negative class (−1). The number of main effects d is chosen from $\{30, 60, 100, 150\}$. The number of underlying nonzero main effects is set to 5 and the number of underlying nonzero interactions is 15. We set $\mu^{(-1)} = \mathbf{0} \in \mathbb{R}^d$ and $\Sigma^{(-1)} = I_d$. We set

$$\mu_j^{(+1)} = \begin{cases} 3 & \text{if } j = 1, \ldots, 5 \\ 0 & \text{if } j = 6, \ldots, d \end{cases},$$

and

$$\Sigma_{i,j}^{(+1)} = \begin{cases} 0.15 & \text{if } (i,j) \text{ or } (j,i) \in \Omega \\ \Sigma_{i,j}^{(-1)} & \text{otherwise} \end{cases},$$

where $\Omega \subset \{1,\ldots,5\} \times \{6,\ldots,d\}$ is the index set of the 15 nonzero interactions. We conduct 1000 simulations and show the performance of discovering nonzero interactions, *i.e.,* the portion of nonzero interactions discovered. The experimental results are shown in Figure 2.3. We observe that, when the weak hierarchy holds between

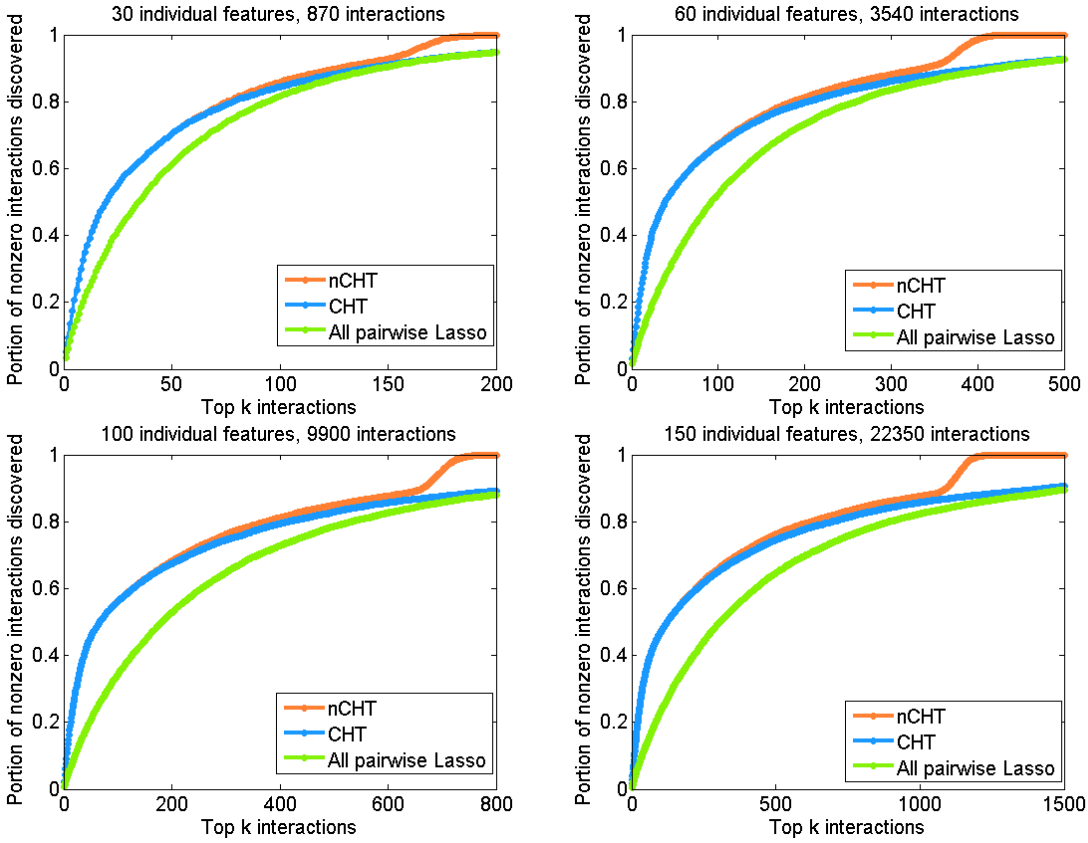

Figure 2.3: Comparison of nCHT, CHT and all pairwise Lasso in terms of recovering underlying nonzero interactions based on test statistic $\lambda_{\max}$.

main effects and interactions, the hierarchical frameworks significantly outperform the all pair Lasso method, which is consistent with the observations in (Bien *et al.*, 2015). The non-convex hierarchical testing method has the same recovery perfor-

mance as the convex one for large thresholds but it can discover more underlying nonzero interactions when the threshold is small.

Chapter 3

DYADIC POSITIVE UNLABELED LEARNING

3.1 Introduction

Labeling unknown data pairs as negative may not be appropriate in the real world. For example, an unknown drug-disease pair does not mean the drug cannot be used to treat the disease. It is just not validated or tested, because there is a huge number of distinct drug-disease pairs. We have the same problem for DDI prediction, where in most cases only positive DDIs can be detected.

To address these challenges we propose a general learning framework called *Dyadic Positive-Unlabeled* (DyPU) learning. The basic setting for DyPU is as follows: we are given a set of data pairs and there is a binary label associated with each pair, some of which are known as positive (+1) with the rest unknown; how can we make use of this small portion of positive data pairs together with the rest unlabeled data pairs to identify positive pairs from the unlabeled data pairs. Inspired by the "ranking at the top" problem (Li *et al.*, 2014b; Agarwal, 2011; Boyd *et al.*, 2012), we introduce a scoring function that assigns ranking scores to each data pair. The ranking scores of positive pairs are required to be higher than those of negative pairs. Different from the classic binary classification or bipartite ranking problem where binary labels or pairwise relations are known, we develop a novel model that enables detecting positive data interactions in positive-only and unlabeled settings by forcing positive pairs rank to "on top of" (i.e., having a higher score than) the average score of the unlabeled pairs. Moreover, our model can make full use of information from the two data points in a data pair that may come from two totally different feature domains.

Our proposed framework is able to incorporate different scoring functions, *e.g.,* the linear function, the sigmoidal function and the rectifier function. When the rectifier function is chosen as the ranking function, the primal optimization problem is hard to solve. We derive the dual formulation of each convex subproblem and show that the associated non-trivial proximal operator of the dual problem admits a closed form solution. We conduct extensive comparison experiments to demonstrate the superiority of the proposed DyPU framework on both drug repositioning and DDI prediction tasks on real world data sets. Before introducing the proposed framework, we will briefly review the related work.

3.1.1 PU Learning

Positive-Unlabeled (PU) Learning (Liu *et al.*, 2003; Liu, 2007; Elkan and Noto, 2008) refers to a set of learning problems based on positive data points and unlabeled data points. Since there are an unknown portion of positive data points unobserved, directly modelling PU learning problem as a binary classification will lead to highly biased models, which is undesirable. Traditional PU learning approaches can be divided into two categories: the two-step approach and the direct approach (Liu, 2007). The general idea of the two-step approach is to first identify a set of "reliable negatives" from unlabeled data points and then build binary classifiers on positives and those identified "reliable negatives". The direct approach mainly refers to a weighted classifier where larger weights are imposed on positive errors and smaller weights are imposed on unlabeled errors. The weights are tuning parameters which may be impractical in real-world applications since the distribution of unlabeled positives is unknown. Traditional PU learning approaches are not suitable for incorporating information from multiple domains and thus are not applicable for detecting interactions of data points.

Sellamanickam *et al.* (2011) applied pairwise ranking SVM (RSVM) (Joachims, 2005; Chapelle and Keerthi, 2010) for PU learning where the ranking scores of positives are required to be larger than those of unlabeled scores. For two-class scenario, pairwise ranking SVM is closely related to "ranking at the top" approach (Li *et al.*, 2014b; Agarwal, 2011) which maximizes the number of positives ranking above any negative data points, and they show similar performance in empirical studies (Sellamanickam *et al.*, 2011). In the PU learning setting, the pairwise ordering information is not complete, therefore directly applying pairwise ranking approaches may lead to biased models.

3.1.2 Detecting Interaction of Data Points

Atias and Sharan (2011) combined prediction scores from Canonical Correlation Analysis (CCA) and label propagation model to predict associations between drugs and their side effects, which is not able to use multi-domain information. There has been an increasing amount of works on recovering gene-disease associations using network-based algorithms (Wu *et al.*, 2008; Lee *et al.*, 2011; Singh-Blom *et al.*, 2013) which however require new data points to be included in the network and thus are limited for prediction purpose.

Gonen and Kaski (2014) proposed Kernelized Bayesian Matrix Factorization (KBMF) to predict drug-target interactions by making use of the information from multiple domains via kernel methods. The Multiple Similarities Collaborative Matrix Factorization (MSCMF) (Zheng *et al.*, 2013), was proposed for drug-target interaction prediction which approximates the indicator matrix by the product of projection matrices of drug and target similarity matrices. To enable out-of-matrix predictions, the matrix factorization type methods rely on kernel/similarity matrices which may be noisy and of poor quality. Moreover, all matrix factorization type methods are

not suitable for the PU learning setting since they treat the unlabeled data points as negatives which suffers from the same problem as binary classification.

Natarajan and Dhillon (2014) applied an inductive matrix completion method (Jain and Dhillon, 2013) to predict gene-disease associations with a bilinear model incorporating features from both domains. Theoretical analysis of inductive matrix completion for PU learning has been recently provided by (Hsieh *et al.*, 2015). However, inductive matrix completion for predicting positive interactions essentially equals to matrix factorization approaches which may confront the same problems of mistakenly categorizing unobserved positive data interactions as "negatives".

Notations: In DyPU there are two data domains, (e.g., drug and disease for drug repositioning, drug and drug for DDI prediction). Assume there are N_1 data points from the first domain and N_2 data points from the second domain. Let $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{N_1}]^T \in \mathbb{R}^{N_1 \times d_1}$ and $Z = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{N_2}]^T \in \mathbb{R}^{N_2 \times d_2}$ be the data matrices of the two domains respectively. $Y \in \mathbb{R}^{N_1 \times N_2}$ represents the indicator matrix where $Y_{i,j}$ is 1 if there is an interaction between data points $\mathbf{x}_i$ and $\mathbf{z}_j$, and 0 if the interaction between them is not observed. We denote the index set of positive interactions as $\mathcal{P} = \{(i,j)|Y_{i,j} = 1, 1 \leq i \leq N_1, 1 \leq j \leq N_2\}$ and the index set of unobserved interactions as $\mathcal{U} = \{(i,j)|Y_{i,j} = 0, 1 \leq i \leq N_1, 1 \leq j \leq N_2\}$. Let $N_{\mathcal{P}} = |\mathcal{P}|$ and $N_{\mathcal{U}} = |\mathcal{U}|$. $[\cdot]_+$ denotes the rectifier/thresholding function where $[x]_+ = x$ if $u > 0$, and 0 if $u \leq 0$. $\mathbb{I}(\cdot)$ denotes the indicator function where $\mathbb{I}(x) = 1$ if $x > 0$, and 0 if $x \leq 0$.

3.2 Scoring Functions

We consider the pairwise interaction detection where the two data points in each pair may come from the same feature domain or totally different feature domains. In general, we define the real-valued scoring function $f : \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \to \mathbb{R}$ for a data pair

$(\mathbf{x}, \mathbf{z}) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2}$ as:

$$f(\mathbf{x}, \mathbf{z}) = g\left(\mathbf{x}^T W \mathbf{z}\right), \tag{3.1}$$

where $g(\cdot)$ is an arbitrary monotonic non-decreasing function. When $g(\cdot)$ is the identity function, *i.e.,* $g(x) = x$, $f(\mathbf{x}, \mathbf{z})$ is a regular bilinear predictor function of $\mathbf{x}$ and $\mathbf{z}$. Other examples of function $g(\cdot)$ include the sigmoid function $g(x) = \frac{1}{1+\exp(-x)}$, the rectifier function $g(x) = [x]_+$ etc. When the data pairs come from the same feature domain, we require the coefficient matrix W to be symmetric, *i.e.,* $W = W^T$, as a non-symmetric W in (3.1) will lead to inconsistent predictions, *i.e.,* $f(\mathbf{x}_i, \mathbf{x}_j) \neq f(\mathbf{x}_j, \mathbf{x}_i)$.

Compared with other approaches such as similarity-based methods, the modelling of a scoring function enables the "cold-start" type prediction, which means the prediction is not dependent on any existing training data. Moreover, compared with factorization type approaches, the bilinear model is also able to produce interpretable results. To see this, we can rewrite the scoring function as

$$f(\mathbf{x}, \mathbf{z}) = g\left(\sum_{i,j} W_{i,j} x_i z_j\right), \tag{3.2}$$

which is essentially a function of the linear combination of feature interactions $x_i z_j$. In other words, the scores assigned to data pairs are determined by their feature interactions. For example, if the coefficient matrix W is sparse, *i.e.,* only a small number of $W_{i,j}$'s are non-zero, then one may identify relevant feature interactions for detecting data interactions.

3.3 Proposed Framework for Positive Interaction Detection

Since an unknown amount of positives are mixed together with unlabeled instances in PU learning, instead of forcing each positive to scores at the absolute top as commonly done in classification/ranking at the top problem studied in (Li *et al.*, 2014b; Boyd *et al.*, 2012; Rudin and Schapire, 2009; Agarwal, 2011), we propose a

novel loss formulation that maximizes the number of positives ranking higher than the average score of unlabeled samples.

Then, our proposed optimization problem is:

$$\min_{W} \frac{1}{N_{\mathcal{P}}} \sum_{(i,j)\in\mathcal{P}} \mathbb{I}\left(\frac{1}{N_{\mathcal{U}}} \sum_{(k,l)\in\mathcal{U}} g\left(\mathbf{x}_k^T W \mathbf{z}_l\right) - g\left(\mathbf{x}_i^T W \mathbf{z}_j\right)\right), \tag{3.3}$$

which aims to minimize the fraction of positive interactions ranked below the average score of unobserved interactions. Note that the proposed problem (3.3) is completely different from existing methodologies for data interaction detection since it does not impose any assumption on the data distribution of unobserved interactions. The objective function in (3.3) is discontinuous and non-convex, which makes the optimization problem difficult to solve. Therefore, we propose to minimize the following convex problem by replacing the indicator function with its convex surrogate:

$$\min_{W} \frac{1}{N_{\mathcal{P}}} \sum_{(i,j)\in\mathcal{P}} \ell\left(\frac{1}{N_{\mathcal{U}}} \sum_{(k,l)\in\mathcal{U}} g\left(\mathbf{x}_k^T W \mathbf{z}_l\right) - g\left(\mathbf{x}_i^T W \mathbf{z}_j\right)\right) := \mathfrak{L}(W), \tag{3.4}$$

where $\ell(\cdot)$ is a convex loss function that is non-decreasing and differentiable. Candidates of such loss functions include the truncated quadratic loss function $\ell(x) = [1+x]_+^2$, the exponential function $\ell(x) = \exp(x)$, the logistic loss function $\ell(x) = \log(1+\exp(x))$ and so on. To prevent the overfitting problem, we solve the following regularized problem instead of directly minimizing (3.4):

$$\min_{W} \quad \mathfrak{L}(W) + \lambda \mathfrak{R}(W) \tag{3.5}$$

where $\mathfrak{R}(W)$ is a regularizer imposed on W. Typically, we assume $\mathfrak{R}(\cdot)$ is convex (not necessarily smooth). Common choices of $\mathfrak{R}(\cdot)$ include the squared Frobenius norm $\|\cdot\|_F^2$ which is equal to the summation of square of entries in the matrix W, the trace norm $\|\cdot\|_*$ which is defined as the summation of singular values of the matrix W, and the matrix l_1 norm which is the summation of the absolute value of matrix entries

etc. Imposing trace norm as the regularizer leads to a low-rank solution where only a small number of underlying latent factors are assumed to contribute to the model. Using the squared Frobenius norm as the regularizer, each element of the solution is required not to be too large. The matrix l_1 norm penalty results in a sparse and interpretable solution which is suitable for large dimensional data.

3.3.1 General Optimization Methods

In general, the proposed framework DyPU considers the case when the problem is for detecting interactions between data points coming from two different feature domains. The convexity of the objective is dependent on the choice of the scoring function. If the scoring function is chosen as a bilinear function, *i.e.,* $f(\mathbf{x}, \mathbf{z}) = \mathbf{x}^T W \mathbf{z}$, then the objective (3.5) is convex, which consists of a convex smooth loss function and a convex regularizer (may or may not be smooth). If the regularizer is one of trace norm, Frobenius norm and the matrix l_1 norm, problem (3.5) can be efficiently solved by well-known optimization methods such as accelerate gradient descent (ACG) (Nesterov, 2004, 1983) and Alternating Direction Method of Multipliers (ADMM) (Boyd *et al.*, 2011). In particular, when the trace norm is used for regularization, one may assume that the coefficient matrix W can be factorized into the product of two low-rank matrices $U \in \mathbb{R}^{d_1 \times r}$ and $V \in \mathbb{R}^{d_2 \times r}$ where $r \ll \min(d_1, d_2)$, i.e., $W = UV^T$. Moreover, the trace norm of W can be equivalently defined as (Fazel *et al.*, 2001; Srebro *et al.*, 2004):

$$\|W\|_* = \min_{W=UV^T} \frac{1}{2} \left(\|U\|_F^2 + \|V\|_F^2 \right).$$

Thus we can solve the problem by alternately minimizing U with V fixed, and minimizing V with U fixed. When the sample interaction detection is considered within only one feature domain, a symmetric prediction is required as discussed in subsection 3.2. The optimization problem with the symmetric constraint $W = W^T$ can be solved

via ADMM (Boyd *et al.*, 2011). For regularizers such as trace norm, l_1 norm, one can adopt the proximal splitting methods (Sra, 2011) to efficiently solve the non-convex problem, which is guaranteed to achieve convergence.

3.4 Dual Formulation with the Rectifier Scoring Function

When the rectifier function $[\cdot]_+$ is chosen as the scoring function, the empirical loss in (3.4) would become

$$\frac{1}{N_{\mathcal{P}}} \sum_{(i,j)\in\mathcal{P}} \ell \left(\frac{1}{N_{\mathcal{U}}} \sum_{(k,l)\in\mathcal{U}} \left[\mathbf{x}_k^T W \mathbf{z}_l\right]_+ - \left[\mathbf{x}_i^T W \mathbf{z}_j\right]_+ \right). \tag{3.6}$$

Note that in (3.6), the rectifier function also truncates the scores of positive data pairs, which leads to a smaller loss when the bilinear score function makes a negative score for a positive data pair and thus essentially weakens the requirement of "positive ranking at the top" and results in a solution of poor quality. To resolve this issue, we remove the max operator in $[\cdot]_+$ for positive data pairs in the objective and thus obtain the following empirical loss:

$$\frac{1}{N_{\mathcal{P}}} \sum_{(i,j)\in\mathcal{P}} \ell \left(\frac{1}{N_{\mathcal{U}}} \sum_{(k,l)\in\mathcal{U}} \left[\mathbf{x}_k^T W \mathbf{z}_l\right]_+ - \mathbf{x}_i^T W \mathbf{z}_j \right). \tag{3.7}$$

When the squared Frobenius norm is used as the regularizer, we have

$$\min_W \frac{1}{N_{\mathcal{P}}} \sum_{(i,j)\in\mathcal{P}} \ell \left(\frac{1}{N_{\mathcal{U}}} \sum_{(k,l)\in\mathcal{U}} \left[\mathbf{x}_k^T W \mathbf{z}_l\right]_+ - \mathbf{x}_i^T W \mathbf{z}_j \right) + \frac{\lambda}{2} \|W\|_F^2. \tag{3.8}$$

The loss part in objective (3.8) is convex because it is the composite of two convex functions. However, the empirical loss function is non-smooth and thus difficult to solve. To overcome the difficulty brought by the rectifier function, we derive the dual form of (3.8) when the quadratic truncated function $[\cdot]_+^2$ is chosen for loss function, which is stated in the following theorem.

Theorem 3.4.1. *Let*

$$\ell(\mathbf{a}) = \max(1 + \mathbf{a}, 0)^2.$$

Then the dual form of problem (3.8) *is*

$$\min_{(\boldsymbol{\alpha},\boldsymbol{\eta})\in\Omega} \frac{1}{2\lambda} \left\| \frac{1}{N_{\mathcal{P}}} \sum_{(i,j)\in\mathcal{P}} \alpha_{(i,j)} \mathbf{x}_i \mathbf{z}_j^T - \sum_{(k,l)\in\mathcal{U}} \eta_{(k,l)} \mathbf{x}_k \mathbf{z}_l^T \right\|_F^2 + \frac{1}{N_{\mathcal{P}}} \sum_{(i,j)\in\mathcal{P}} \ell_* \left(\alpha_{(i,j)}\right). \quad (3.9)$$

where $\alpha_{(i,j)}$*'s and* $\eta_{(k,l)}$*'s are dual variables associated with positive data pairs* $(i,j) \in \mathcal{P}$ *and unlabeled sample pairs* $(k,l) \in \mathcal{U}$ *respectively;* $\ell_*(\cdot)$ *is the conjugate function of* $\ell(\cdot)$ *defined as follows:*

$$\ell_*(\beta) = \sup_u \{u\beta - l(u)\} = -\beta + \frac{\beta^2}{4},\ \beta \geq 0;$$

the domain Ω *is defined as*

$$\Omega = \Big\{ \boldsymbol{\alpha} \in \mathbb{R}^{N_{\mathcal{P}}}, \boldsymbol{\eta} \in \mathbb{R}^{N_{\mathcal{U}}} : \alpha_{(i,j)} \geq 0,\ \textit{for } (i,j) \in \mathcal{P}$$
$$0 \leq \eta_{(k,l)} \leq \frac{1}{N_{\mathcal{P}} N_{\mathcal{U}}} \sum_{(i,j)\in\mathcal{P}} \alpha_{(i,j)},\ \textit{for } (k,l) \in \mathcal{U} \Big\},$$

where elements of $\boldsymbol{\alpha}$ *and* $\boldsymbol{\eta}$ *correspond to* $\alpha_{(i,j)}$*'s and* $\eta_{(k,l)}$*'s, respectively. Let* $\boldsymbol{\alpha}^*$ *and* $\boldsymbol{\eta}^*$ *be the optimal solution to the dual problem* (3.9)*. Then, the optimal solution* W^* *to the primal problem* (3.8) *is given by*

$$W^* = \frac{1}{\lambda} \left(\frac{1}{N_{\mathcal{P}}} \sum_{(i,j)\in\mathcal{P}} \alpha^*_{(i,j)} \mathbf{x}_i \mathbf{z}_j^T - \sum_{(k,l)\in\mathcal{U}} \eta^*_{(k,l)} \mathbf{x}_k \mathbf{z}_l^T \right). \quad (3.10)$$

Proof. It can be verified that the conjugate function of $\ell(\cdot)$ is

$$\ell_*(\beta) = \sup_u \{u\beta - \ell(u)\} = -\beta + \frac{\beta^2}{4},\ \beta \geq 0.$$

Since $\ell(\cdot)$ is convex and closed, we can rewrite it in terms of its convex conjugate form, *i.e.,*

$$\ell(u) = \max_{\beta\geq 0} \beta u - \ell_*(\beta).$$

Thus, the formulation

$$\min_{W} \frac{1}{N_{\mathcal{P}}} \sum_{(i,j)\in\mathcal{P}} \ell\left(\frac{1}{N_{\mathcal{U}}} \sum_{(k,l)\in\mathcal{U}} \left[\mathbf{x}_k^T W \mathbf{z}_l\right]_+ - \mathbf{x}_i^T W \mathbf{z}_j\right) + \frac{\lambda}{2}\|W\|_F^2 \tag{3.11}$$

can be rewritten as

$$\begin{aligned}\min_{W} \max_{\boldsymbol{\alpha}\geq 0} \frac{1}{N_{\mathcal{P}}} \sum_{(i,j)\in\mathcal{P}} \alpha_{(i,j)} \left(\frac{1}{N_{\mathcal{U}}} \sum_{(k,l)\in\mathcal{U}} \left[\mathbf{x}_k^T W \mathbf{z}_l\right]_+ - \mathbf{x}_i^T W \mathbf{z}_j\right) \\ - \frac{1}{N_{\mathcal{P}}} \sum_{(i,j)\in\mathcal{P}} \ell_*\left(\alpha_{(i,j)}\right) + \frac{\lambda}{2}\|W\|_F^2.\end{aligned} \tag{3.12}$$

Since problem (3.12) is convex in W and concave in $\boldsymbol{\alpha}$ and its feasible domain is convex, the strong max-min property is satisfied (Boyd and Vandenberghe, 2004). Hence, we swap the min and max operator and obtain

$$\begin{aligned}\max_{\alpha\geq 0} \min_{W} \frac{1}{N_{\mathcal{P}}} \sum_{(i,j)\in\mathcal{P}} \alpha_{(i,j)} \left(\frac{1}{N_{\mathcal{U}}} \sum_{(k,l)\in\mathcal{U}} \left[\mathbf{x}_k^T W \mathbf{z}_l\right] - \mathbf{x}_i^T W \mathbf{z}_j\right) \\ - \frac{1}{N_{\mathcal{P}}} \sum_{(i,j)\in\mathcal{P}} \ell_*\left(\alpha_{(i,j)}\right) + \frac{\lambda}{2}\|W\|_F^2.\end{aligned} \tag{3.13}$$

We first consider the minimization problem, i.e.,

$$\min_{W} \frac{1}{N_{\mathcal{P}}} \sum_{(i,j)\in\mathcal{P}} \alpha_{(i,j)} \left(\frac{1}{N_{\mathcal{U}}} \sum_{(k,l)\in\mathcal{U}} \left[\mathbf{x}_k^T W \mathbf{z}_l\right]_+ - \mathbf{x}_i^T W \mathbf{z}_j\right) + \frac{\lambda}{2}\|W\|_F^2, \tag{3.14}$$

where we omit the term $-\frac{1}{N_{\mathcal{P}}} \sum_{(i,j)\in\mathcal{P}} \ell_*(\alpha_{(i,j)})$ which is constant with respect to W.

To handle the max operator in $[\cdot]_+$, we introduce slack variables $\xi_{(k,l)} \geq 0, (k,l) \in \mathcal{U}$ for the $N_{\mathcal{U}}$ scores of unlabeled data pairs, and rewrite formulation (3.14) as

$$\begin{aligned}\min_{W} \frac{1}{N_{\mathcal{P}}} \sum_{(i,j)\in\mathcal{P}} \alpha_{(i,j)} \left(\frac{1}{N_{\mathcal{U}}} \sum_{(k,l)\in\mathcal{U}} \xi_{(k,l)} - \mathbf{x}_i^T W \mathbf{z}_j\right) + \frac{\lambda}{2}\|W\|_F^2, \\ \text{s.t.} \quad \mathbf{x}_k^T W \mathbf{z}_l \leq \xi_{(k,l)}, \quad \xi_{(k,l)} \geq 0, \quad (k,l) \in \mathcal{U}.\end{aligned} \tag{3.15}$$

The Lagrangian function of problem (3.15), $\mathcal{L}(W, \boldsymbol{\xi}, \boldsymbol{\eta}, \boldsymbol{\gamma})$, is

$$\begin{aligned} &\frac{1}{N_{\mathcal{P}}} \sum_{(i,j)\in\mathcal{P}} \alpha_{(i,j)} \left(\frac{1}{N_{\mathcal{U}}} \sum_{(k,l)\in\mathcal{U}} \xi_{(k,l)} - \mathbf{x}_i^T W \mathbf{z}_j \right) + \frac{\lambda}{2} \|W\|_F^2 \\ &+ \sum_{(k,l)\in\mathcal{U}} \eta_{(k,l)} \left(\mathbf{x}_k^T W \mathbf{z}_l - \xi_{(k,l)} \right) - \sum_{(k,l)} \gamma_{(k,l)} \xi_{(k,l)}. \end{aligned} \tag{3.16}$$

Then the dual problem associated with (3.15) is

$$\max_{\boldsymbol{\eta},\boldsymbol{\gamma}\geq 0} \min_{W,\boldsymbol{\xi}} \mathcal{L}(W, \boldsymbol{\xi}, \boldsymbol{\eta}, \boldsymbol{\gamma}). \tag{3.17}$$

By deriving the optimality conditions for minimizing $\mathcal{L}(W, \boldsymbol{\xi}, \boldsymbol{\eta}, \boldsymbol{\gamma})$ with respect to W and $\boldsymbol{\xi}$, we obtain

$$\begin{cases} W = \dfrac{1}{\lambda} \left(\dfrac{1}{N_{\mathcal{P}}} \displaystyle\sum_{(i,j)\in\mathcal{P}} \alpha_{(i,j)} \mathbf{x}_i \mathbf{z}_j^T - \sum_{(k,l)\in\mathcal{U}} \eta_{(k,l)} \mathbf{x}_k \mathbf{z}_l^T \right) \\ 0 \leq \eta_{(k,l)} \leq \dfrac{1}{N_{\mathcal{P}} N_{\mathcal{U}}} \displaystyle\sum_{(i,j)\in\mathcal{P}} \alpha_{(i,j)} \\ \gamma_{(k,l)} = \dfrac{1}{N_{\mathcal{P}} N_{\mathcal{U}}} \displaystyle\sum_{(i,j)\in\mathcal{P}} \alpha_{(i,j)} - \eta_{(k,l)} \end{cases} \tag{3.18}$$

By plugging the last equation of (3.18) into (3.17), we obtain

$$\max_{\boldsymbol{\eta}\geq 0} \min_{W} \; -\frac{1}{N_{\mathcal{P}}} \sum_{(i,j)\in\mathcal{P}} \alpha_{(i,j)} \mathbf{x}_i^T W \mathbf{z}_j + \sum_{(k,l)\in\mathcal{U}} \eta_{(k,l)} \mathbf{x}_k^T W \mathbf{z}_l + \frac{\lambda}{2} \|W\|_F^2. \tag{3.19}$$

Considering the first equation in (3.18), we obtain that the objective function in (3.19) can be written as

$$\begin{aligned} &\operatorname{tr} \left(- \left(\frac{1}{N_{\mathcal{P}}} \sum_{(i,j)\in\mathcal{P}} \alpha_{(i,j)} \mathbf{z}_j \mathbf{x}_i^T - \sum_{(k,l)\in\mathcal{U}} \eta_{(k,l)} \mathbf{z}_l \mathbf{x}_k^T \right)^T W \right) + \frac{\lambda}{2} \|W\|_F^2 \\ =& -\lambda \|W\|_F^2 + \frac{\lambda}{2} \|W\|_F^2 = -\frac{\lambda}{2} \|W\|_F^2. \end{aligned} \tag{3.20}$$

Plugging the first equation in (3.18) into problem (3.20), the dual problem (3.17) thus

can be written as

$$\begin{aligned}
\max_{\boldsymbol{\eta},\boldsymbol{\alpha}} \; & -\frac{1}{2\lambda}\left\|\frac{1}{N_{\mathcal{P}}}\sum_{(i,j)\in\mathcal{P}}\alpha_{(i,j)}\mathbf{x}_i\mathbf{z}_j^T-\sum_{(k,l)\in\mathcal{U}}\eta_{(k,l)}\mathbf{x}_k\mathbf{z}_l^T\right\|_F^2-\frac{1}{N_{\mathcal{P}}}\sum_{(i,j)\in\mathcal{P}}\ell_*(\alpha_{(i,j)}) \\
& := \mathfrak{D}(\boldsymbol{\alpha},\boldsymbol{\eta}) \\
\text{s.t. } & \alpha_{(i,j)}\geq 0,\ (i,j)\in\mathcal{P} \\
& 0\leq\eta_{(k,l)}\leq\frac{1}{N_{\mathcal{P}}N_{\mathcal{U}}}\sum_{(i,j)\in\mathcal{P}}\alpha_{(i,j)},\quad (k,l)\in\mathcal{U}
\end{aligned} \tag{3.21}$$

Changing the negative signs and replacing the max operator with min operator completes the proof. □

We observe that the objective function of the dual problem (3.9) is smooth. Thus (3.9) can be efficiently solved by proximal (projected) gradient methods (Nesterov, 2004, 1983; Ji and Ye, 2009; Beck and Teboulle, 2009; Gong *et al.*, 2013a; Wright *et al.*, 2009) which were demonstrated to be very efficient for solving regularized (constrained) optimization problems. A critical step for proximal (projected) gradient methods is to compute the proximal operator (projection) problem associated with the constraints. We next show that the non-trivial proximal operator (projection) problem [1] associated with problem (3.9) admits a closed form solution.

3.4.1 Efficient Algorithm for Computing the Proximal Operator

We propose to use proximal algorithms to solve problem (3.9) which computes a sequence of proximal operators. For clearer presentation, we simplify the notations and write the proximal operator problem associated with problem (3.9) at the k-th

[1] For a constrained optimization problem, the subproblem is a projection problem which can be viewed as a special case of the proximal operator problem. Thus, we only mention the proximal operator problem in the sequel.

step as

$$\begin{aligned} \min_{\boldsymbol{\eta},\boldsymbol{\alpha}} \ & \frac{1}{2}\left\|\boldsymbol{\eta}-\mathbf{u}\right\|_2^2+\frac{1}{2}\left\|\boldsymbol{\alpha}-\mathbf{v}\right\|_2^2 \\ \text{s.t. } & 0\leq\eta_j\leq\frac{1}{mn}\sum_i\alpha_i, \qquad j=1,\ldots,n \\ & \alpha_i\geq 0, \qquad\qquad\qquad\quad i=1,\ldots,m \end{aligned} \tag{3.22}$$

where n represents $N_{\mathcal{U}}$, m represents $N_{\mathcal{P}}$, dual variables $\boldsymbol{\eta}\in\mathbb{R}^n$ and $\boldsymbol{\alpha}\in\mathbb{R}^m$, and

$$\begin{aligned} \mathbf{u} &= \boldsymbol{\eta}-t^{(k)}\nabla_{\boldsymbol{\eta}}\mathfrak{D}(\boldsymbol{\alpha},\boldsymbol{\eta}) \\ \mathbf{v} &= \boldsymbol{\alpha}-t^{(k)}\nabla_{\boldsymbol{\alpha}}\mathfrak{D}(\boldsymbol{\alpha},\boldsymbol{\eta}) \end{aligned}, \tag{3.23}$$

$t^{(k)}$ is the stepsize. The proximal operator problem (3.22) is highly non-trivial because of the affine constraints on dual variables. For problem (3.22), we have the following property:

Theorem 3.4.2. *Let* $\kappa:=\frac{1}{mn}\sum_{i=1}^m\alpha_i$. *Then problem* (3.22) *is equivalent to minimizing the following function:*

$$q(\kappa,s)=\frac{1}{2}\sum_{j=1}^n[u_j-\kappa]_+^2+\frac{1}{2s}\left(\sum_{i=1}^s v_i-mn\kappa\right)^2+\frac{1}{2}\sum_{i=s+1}^m v_i^2, \tag{3.24}$$

where

$$s=\max\left\{s\in\{1,\ldots,m\}:v_s-\frac{1}{s}\left(\sum_{i=1}^s v_i-mn\kappa\right)>0\right\}.$$

Proof. Notice that $0\leq\eta_j\leq\kappa=\frac{1}{mn}\sum_{i=1}^m\alpha_i$. Thus, we have

$$\min_{\boldsymbol{\eta}}\frac{1}{2}\left\|\boldsymbol{\eta}-\mathbf{u}\right\|_2^2=\frac{1}{2}\sum_{j=1}^n[u_j-\kappa]_+^2+\frac{1}{2}\sum_{j=1}^n[-u_j]_+^2. \tag{3.25}$$

We next focus on the following problem:

$$\begin{aligned} \min_{\alpha} \ & \frac{1}{2}\|\boldsymbol{\alpha}-\mathbf{v}\|_2^2 \\ \text{s.t. } & \frac{1}{mn}\sum_{i=1}^m\alpha_i=\kappa, \\ & \alpha_i\geq 0,\ i=1,\ldots,m. \end{aligned} \tag{3.26}$$

Without loss of generality, we assume that $v_i, i = 1, \ldots, m$ are sorted in descending order. It has been shown in (Duchi *et al.*, 2008) that problem (3.26) admits the following closed-form solution:

$$\alpha_i = \max(v_i - \gamma, 0), \tag{3.27}$$

where

$$\gamma = \frac{1}{s}\left(\sum_{i=1}^{s} v_i - mn\kappa\right),$$

and

$$s = \max\left\{s \in \{1, \ldots, m\} : v_s - \frac{1}{s}\left(\sum_{i=1}^{s} v_i - mn\kappa\right) > 0\right\}. \tag{3.28}$$

Plugging (3.27) into (3.26), we obtain that the optimum objective value of (3.26) is

$$\frac{1}{2s}\left(\sum_{i=1}^{s} v_i - mn\kappa\right)^2 + \frac{1}{2}\sum_{i=s+1}^{m} v_i^2, \tag{3.29}$$

which together with (3.25) and (3.28) implies the conclusion. □

Based on the above theorem, we can transform the proximal operator problem in (3.22) to the following optimization problem:

$$\begin{aligned} &\min_{\kappa, s} \; q(\kappa, s) \\ &\text{s.t. } s = \max\left\{s \in \{1, \ldots, m\} : v_s - \frac{1}{s}\left(\sum_{i=1}^{s} v_i - mn\kappa\right) > 0\right\}, \end{aligned} \tag{3.30}$$

where constant items are omitted. The constraint in the above optimization problem can be rewritten as

$$\begin{aligned} s &= \{s \in \{1, \ldots, m-1\} : h(s) < \kappa \leq h(s+1)\} \\ \text{where } h(s) &= \frac{1}{mn}\left(\sum_{i=1}^{s} v_i - sv_s\right), \end{aligned} \tag{3.31}$$

which immediately indicates that $q(\kappa, s)$ is a piecewise quadratic function with respect to κ and the points where it changes from one quadratic function to another one are

included in the following set:

$$\mathcal{C} = \{u_1, \cdots, u_n, h(1), \cdots, h(m)\}. \tag{3.32}$$

By sorting the entries of $\mathcal{C}$ in ascending order, we know that for any adjacent entries c_ℓ and $c_{\ell+1}$ ($\ell = 1, \cdots, m+n-1$), $q(\kappa, \ell)$ is quadratic with respect to κ in the interval $[c_\ell, c_{\ell+1}]$. Thus, it is easy to obtain a minimum solution of $q(\kappa, \ell)$ in the interval $[c_\ell, c_{\ell+1}]$, that is

$$\kappa_\ell^\star = \arg\min_\kappa q(\kappa, \ell), \text{ s.t. } \kappa \in [c_\ell, c_{\ell+1}].$$

Therefore, the global solution of $q(\kappa, s)$ is

$$(\kappa^\star, s^\star) = \arg\min_{\kappa, s} q(\kappa, s), \text{ s.t. } \kappa = \kappa_s^\star, \ s = 1, \cdots, m+n-1.$$

Thus, the optimal solution to (3.22) can be written as

$$\alpha_i^\star = \max\left(v_i - \gamma^\star, 0\right), \ i = 1, \cdots, n,$$

$$\eta_j^\star = \min\left(\kappa^\star, \max\left(u_j, 0\right)\right), \ j = 1, \cdots, m.$$

where

$$\gamma^\star = \frac{1}{s^\star}\left(\sum_{i=1}^{s^\star} v_i - mn\kappa^\star\right).$$

One may adopt the accelerate gradient descent (ACG) algorithm (Nesterov, 2004, 1983; Ji and Ye, 2009; Beck and Teboulle, 2009) to solve the optimization problem, which enjoys a convergence rate of $O(1/k^2)$. The computation of solving problem (2.10) takes $O(N_\mathcal{P} + N_\mathcal{U})$ and thus the time complexity of computing the proximal operator (2.8) is dominated by the sorting step which is $O((N_\mathcal{P} + N_\mathcal{U}) \log(N_\mathcal{P} + N_\mathcal{U}))$.

3.5 Experimental Results of Drug Discovery Problems

3.5.1 Data Description

In the study, we collect 1255 drug molecules from DrugBank (Wishart *et al.*, 2008) and we use chemical structure information as the data features in the experiments. We

use a fingerprint corresponding to the 881 chemical substructures to encode the drug chemical structure. Each drug is represented by an 881-dimensional binary profile whose elements encode for the presence or absence of each PubChem substructure by 1 or 0, respectively. A description of the 881 chemical substructures can be found at the website of PubChem [2] .

We collect known uses of drugs from MEDI database (Wei *et al.*, 2013), which is an ensemble medication indication resource based on multiple commonly used medication resources (e.g., RxNorm, MedlinePlus, and Wikipedia). Indications in MEDI are coded as International Classification of Diseases, 9th edition (ICD9) codes. We group ICD9 codes based on their first 3 digits to avoid trivial predictions (i.e., re-purpose a drug from a disease to very similar diseases). Also, we exclude ill-defined ICD9 groups and rare diseases, and obtain 300 ICD9 groups as diseases in our drug repositioning study. Between our 1255 drugs and 300 diseases, there are 12,493 distinct drug-disease interactions in the dataset. We also construct a disease association matrix based on a real-world Electronic Medical Records (EMR) data warehouse, which includes a longitudinal EMR of 223,091 patients over 4 years. We use the possibility for co-occurrence of two given diseases within a 30-day window in the same individual as the association score between two diseases, and obtain a 300 by 300 matrix to denote disease associations.

We obtain DDIs from DrugBank (Wishart *et al.*, 2008), which are extracted from drug's package inserts, as our known set of DDIs. Among our 922 drugs, there are 9,253 distinct pairwise DDIs in the dataset.

[2]PubChem substructure description. Retrieved February 11, 2016 from $ftp://ftp.ncbi.nlm.nih.gov/pubchem/specifications/pubchem\_fingerprints.txt$

3.5.2 Experiment Settings and Performance Evaluation

In our experiments, we compare our proposed DyPU with rectifier scoring function (DyPU) and other four state-of-the-art approaches including inductive matrix completion (IMC) with logistic loss (Natarajan and Dhillon, 2014; Hsieh *et al.*, 2015), Kernelized Bayesian Matrix Factorization factorization (KBMF) (Gonen and Kaski, 2014), and the two-step PU learning using Naive Bayes Classifiers for both the identification of reliable negatives and classification (Liu *et al.*, 2003). We also include the support vector machine (SVM) as the baseline approach to investigate the performances of classical binary classification. In SVM, we use the stack of the features of each data point in a data pair as the input feature vector. The implementation of KBMF [3] is released by the authors and we use liblinear to implement SVM [4] . The IMC approach is able to use the information from both drugs and diseases and they both make low-rank assumptions. The squared Frobenius norm is used as the regularizer for the proposed DyPU approaches. In our empirical studies, the prediction performance of low-rank approaches is insensitive to the number of ranks and achieves the highest in the range between 5 and 15, thus we only report the results when the rank of coefficient matrix is set as 10.

Since the negatives are completely unknown for PU learning problems, using measurements such as AUC will lead to misleading results which essentially assume unknown labels as negatives and is, however, adopted in most existing literatures. Also, for the detection of positive interactions between data points, it is desired that positives always enjoy higher ranking scores than negatives and a good model is always able to recover more true positives than others. Therefore, besides the F1 score which is a standard evaluation metric used in PU learning, we also use the measurements

[3] http://research.cs.aalto.fi/pml/software/kbmf/

[4] https://www.csie.ntu.edu.tw/ cjlin/liblinear/

from learning to rank to evaluate the performances. Specifically, we adopt the mean precision at top k (mPrec@k), the mean recall at top k (mRecall@k), the mean average precision at top k (mAP@k) and the mean F1 at top k (mF1@k). In real-world drug discovery, only a small number of confident predictions are of interest due to the limitation of resources for clinical validation. Thus, we set the number k as 1 and 3. The reported performances are obtained from 5 rounds of experiments on different splits of data pairs where 50% data pairs are used for training and the rest are used for testing.

A competent model for detecting positive interactions of data points is supposed to be robust to the ratio of observed positive interactions. Also, in practice, it is typically difficult to observe a validated positive sample interactions. Both drug repositioning and DDI prediction tasks need huge clinical efforts to validate. Therefore, a model that is able to make accurate predictions and insensitive to the number of observed positive data pairs is highly desired. Thus, to test the robustness of the proposed model, we randomly conceal $a\%$ of the positive data pairs in the training data by treating them as unobserved. Then we train all the models on the data with $a\%$ of positive data pairs as unlabeled and compare their performances on the testing data. Note that we only flip the labels of positive pairs in the whole training data and keep the testing data the same as before. Therefore, by decreasing the number of observed positive data pairs in training data, we evaluate the robustness of models in terms of recovering true positive interactions of data points in the model. In the experiment, the ratio of positive pairs to be concealed, $a\%$, varies from $\{0\%, 30\%, 60\%, 90\%\}$ where 0% corresponds to the original training data.

3.5.3 Drug Repositioning

In drug repositioning problems, an interaction between a drug and a disease exists if the drug can be used to treat the disease. We compare our proposed methods with competing approaches on detecting such drug-disease interactions. In the experiments, we extract the top 60 principal components from the disease association matrix (described in section 3.5.1) as the latent feature of disease. Since the traditional task of drug repositioning focuses on discovering potential effective drugs for each known disease, we fix all the diseases and conduct validation procedures by randomly splitting drugs. The prediction performance achieved by different methods with different ratios of flipped positive pairs is shown in Figure 3.1. From the figure, we observe that the proposed DyPU and IMC approaches outperform baseline methods including the Two-Step Naive Bayes and SVM significantly indicating the importance of taking advantage of the information of feature interactions. For the original data set or the data set with a small number of concealed positive data pairs, our proposed method achieves comparable prediction performance with IMC. However, as the number of positive pairs decreases and useful information becomes scarcer, the prediction performances of all models decreases in different extents, which is expected. Among all the models, the proposed DyPU with the rectifier function is the most stable model which is not sensitive to the ratio of positive data pairs. The fast decay of performance obtained by the baseline methods validates the hypothesis that mistakenly categorizing unlabeled data pairs as negatives will yield biased models. We also observe that, the traditional two-step PU learning with Naive Bayes Classier achieves a fairly stable performance when the conceal-ratio is 90%, which also indicates the importance of differentiating the settings of PU learning and binary classification.

Besides testing the methods for traditional drug repositioning problem, we are

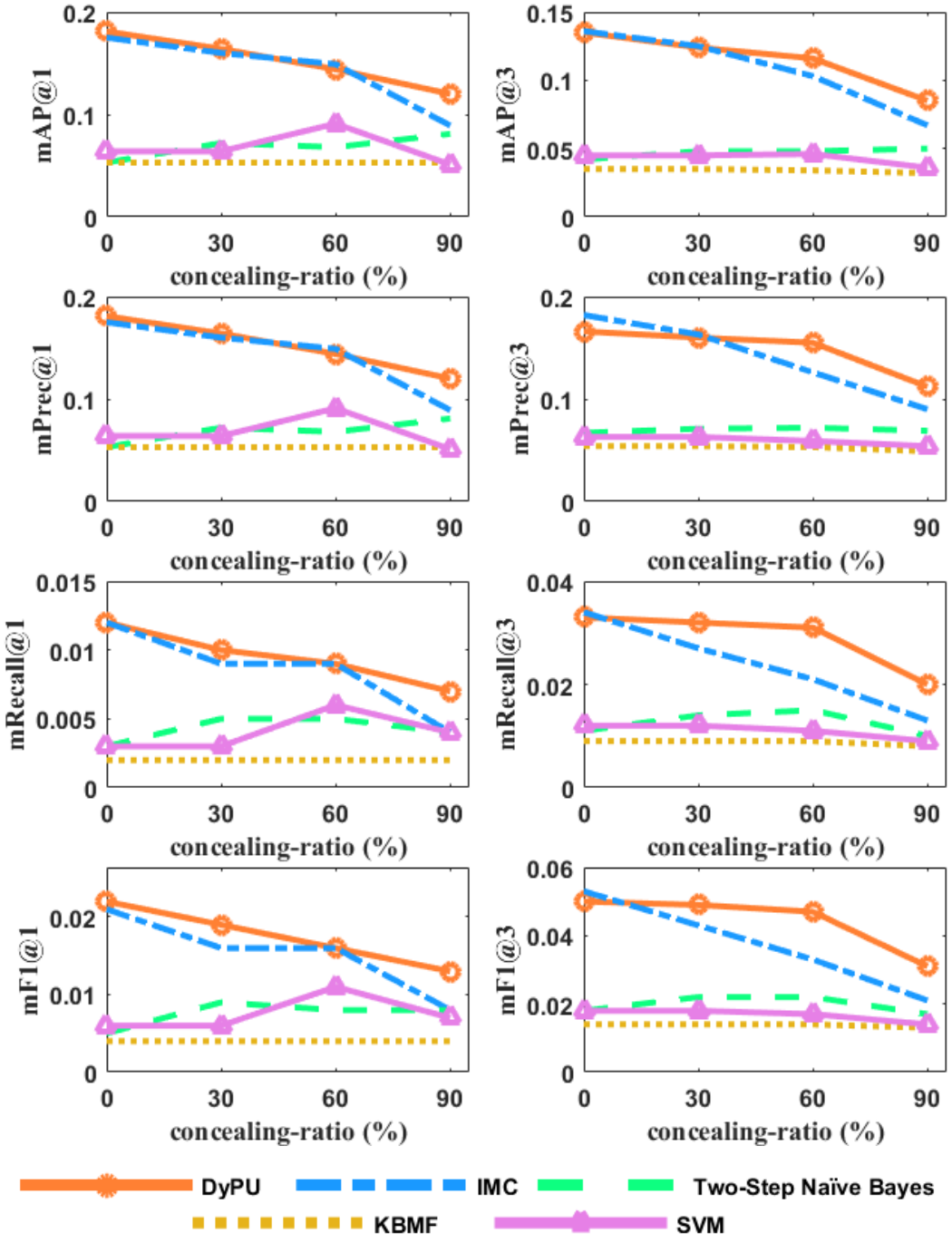

Figure 3.1: Performance comparison of five methods with increasing ratios of concealed positive data pairs on drug repositioning tasks when diseases are observed.

also interested in the ability of predicting the interaction between a new drug and a new disease, which is a much harder problem. In this experiment, we split the data in both drug-wise and disease-wise manner. We train models on the pairs between training drug data points and training disease data points and test the model on the pairs where neither the drug or the disease is seen in the training stage. The validation setting mimics a real-world setting: once rare/unknown diseases without any treatment information arise, a competent drug repositioning method should predict potential new treatments based on characteristics of the new drug molecules and comorbidities of the new diseases. The prediction performance achieved by different methods with different ratios of concealed positive pairs is shown in Figure 3.2.

The performance patterns are very similar to the scenario where the testing diseases are known in the training stage. We observe that the robustness of the proposed DyPU is more remarkable in this setting. For example, even when only 10% of positive interactions are used, the proposed DyPU can still achieve a comparable mean average precision@3 of using 70% of positive data pairs for training while the performance of IMC decays faster than before. Note that, in such case, traditional two-step PU learning method is not applicable.

3.5.4 Drug-Drug Interaction

In this experiment, we use the chemical structure features to predict the positive DDIs. The problem of DDI prediction is restricted to only one domain (i.e., members of a data pair are drugs), therefore traditional approaches are not directly applicable in this scenario. We implement our proposed DyPU with the rectifier scoring function and compare it with the IMC methods (Natarajan and Dhillon, 2014; Jain and Dhillon, 2013). To achieve symmetric predictions, the coefficient matrix W in the models of IMC is factorized into a product of a low-rank matrix U and its transpose

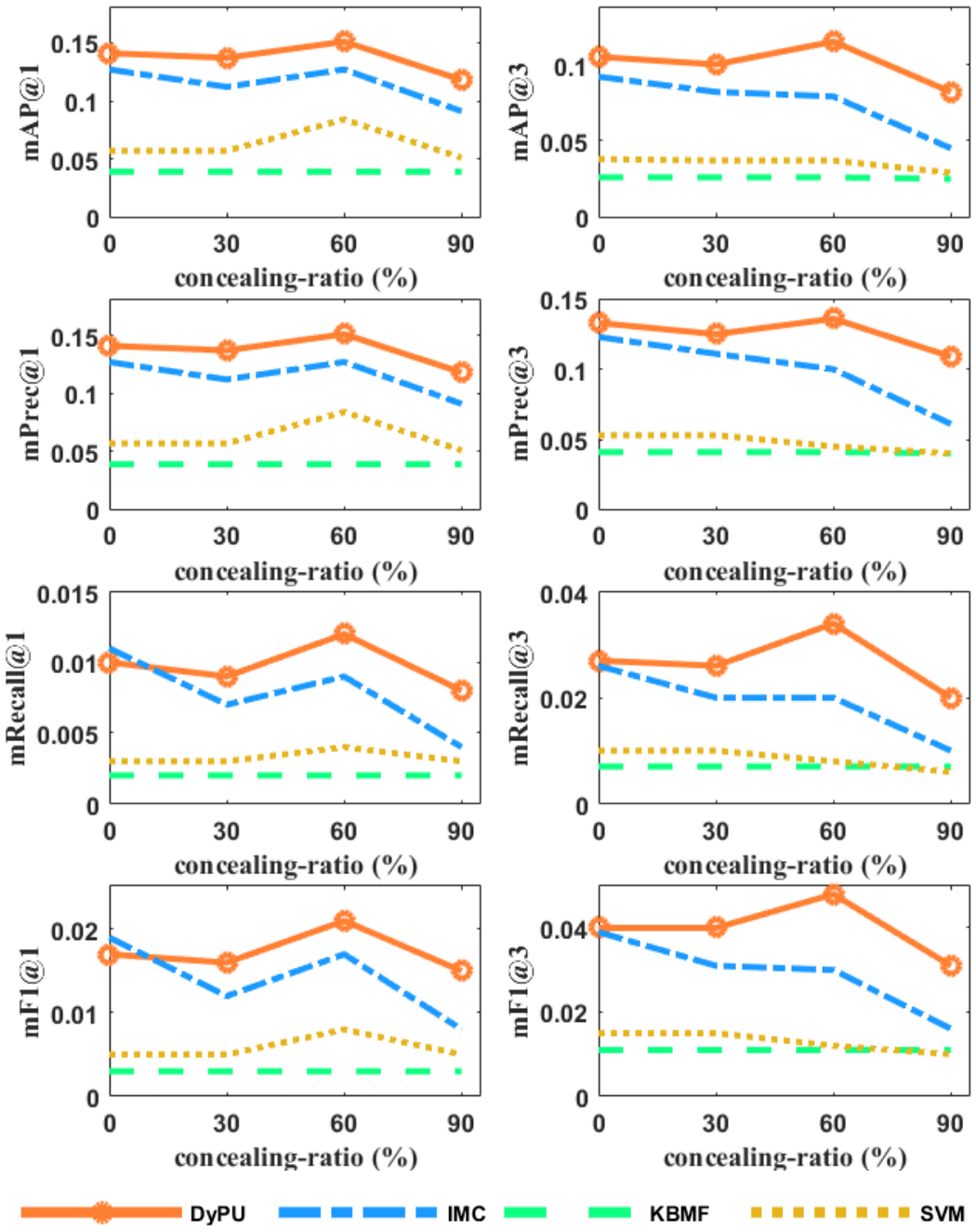

Figure 3.2: Performance comparison of four methods with increasing ratios of concealed positive data pairs on drug repositioning tasks when diseases are unknown.

U^T, *i.e.*, $W = UU^T$, and we solve the formulations with proximal splitting methods.

We are interested in the general predictive power of predicting the interactions

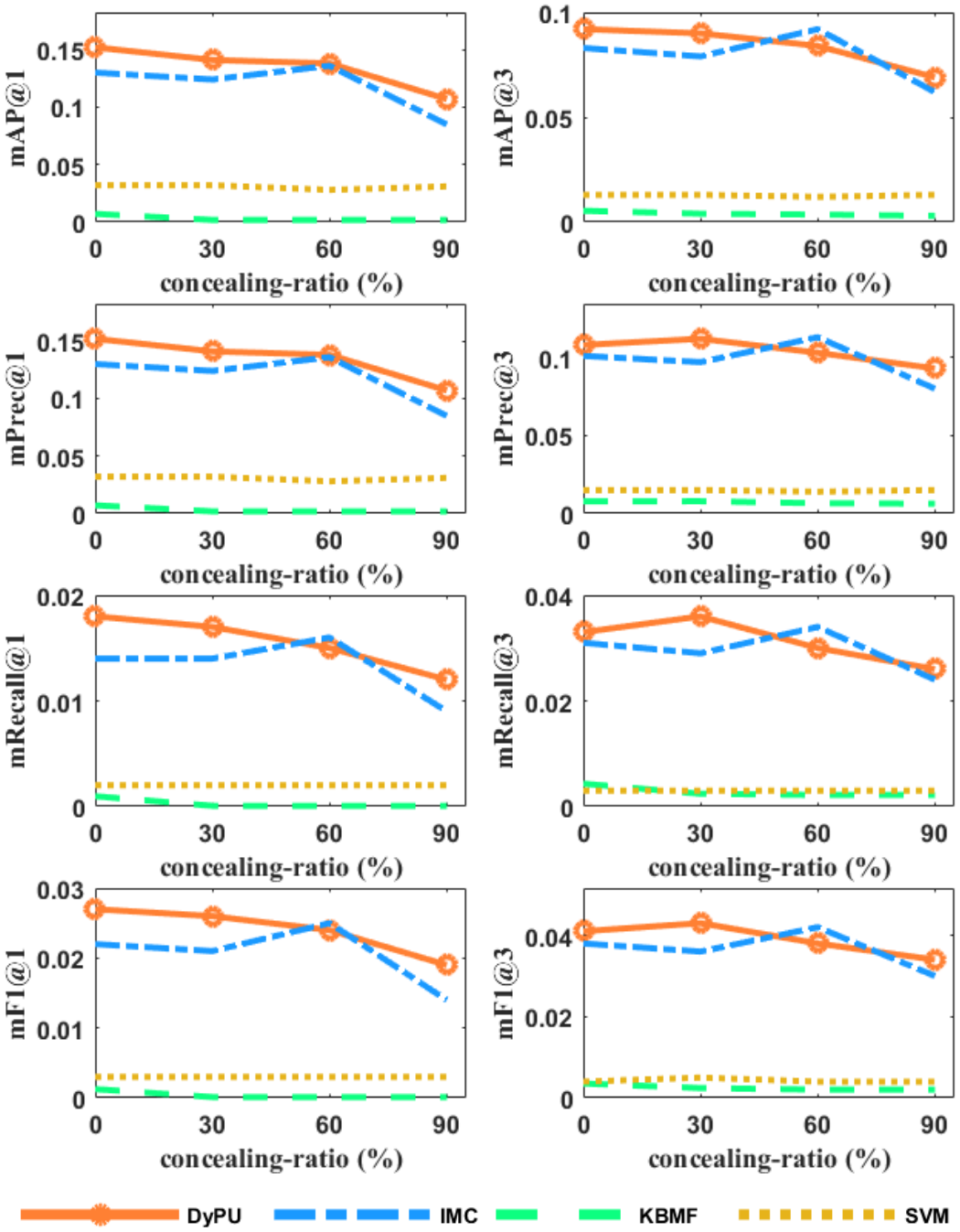

Figure 3.3: Performance comparison of four methods with increasing ratios of concealed positive data pairs on DDI prediction tasks.

between arbitrary drug pairs which could be those of a newly developed drug (i.e., without any known DDI) and existing drugs. We design an experiment to compare the prediction abilities of our proposed framework with the IMC approach, KBMF and SVM. We randomly split the drugs into equally sized training and testing sets and train models based on the data pairs among half of the drugs, and then test their performance of predicting the positive interactions on the remaining half of the data. Similarly, for each drug in the testing set, we compute the average precision@k, precision@k, recall@k and F1@k and obtain the mean values by averaging the scores over all testing drugs. The predictive performance is shown in Figure 3.3, which shows the advantages of our proposed DyPU in DDI prediction. Overall, it outperforms other baseline methods when an increasing number of positive data pairs are concealed, which further demonstrates the superiority of the DyPU in the PU learning setting. The IMC approach exhibits unstable prediction performance for different ratios of concealed positive data pairs, which further demonstrates that treating unlabeled data pairs as negatives will result in biased models. The DyPU with the rectifier function has an overall higher predictive power in terms of detecting positive interactions between drugs, which has the potential to alert the public to possible dangerous DDIs even before a drug officially enters the market.

3.6 Genralizing Ranking Above Average to Ranking Above Average of the Top-ranked

In the framework of DyPU, a model is trained to discover positive data pairs by requiring positve data pairs to rank above the average score of unknown data pairs. Can we find a better representation of unknown data pairs than their average ranking score? An intuitive idea is to use the score of the topmost or the top-ranked unknown data pairs instead of the whole data to form the representation, which is

closely related to top ranking problems in the area of bipartite ranking (Li *et al.*, 2014b; Agarwal, 2011; Tsochantaridis *et al.*, 2005; Boyd *et al.*, 2012; Clémençon and Vayatis, 2007; Christakopoulou and Banerjee, 2005; Burges *et al.*, 2007).

In many applications such as recommendation system, information retrieval and drug discovery, only instances (e.g., data pairs in drug discovery settings stated previously) ranked at the top of the list are of interest due to limited time or resources for further examination. To learn a well-performing ranking model, the classic approach is to use pairwise ranking methods (Joachims, 2002; Freund *et al.*, 2003; Herbrich *et al.*, 1999; Burges *et al.*, 2005) which optimize the preference orders of sample pairs. However, these methods usually cannot guarantee the ranking performance at the top of the ranking list. Moreover, the computation complexity of most pairwise models is quadratic in sample size thus not scalable on large-scale datasets. Recently, increasing attention has been paid to designing methods to optimize the the ranking performance at the top of a list (Li *et al.*, 2014b; Agarwal, 2011; Tsochantaridis *et al.*, 2005; Boyd *et al.*, 2012; Clémençon and Vayatis, 2007; Christakopoulou and Banerjee, 2005; Burges *et al.*, 2007).

The approaches to the above top ranking problem can be roughly categorized into two types. The first type of approach directly optimizes the ranking measurements that emphasize top relevant instances such as average precision (Yue *et al.*, 2007), discounted cumulative gain (DCG) (Cossock and Zhang, 2008; Xu and Li, 2007; Chapelle *et al.*, 2007; Taylor *et al.*, 2008; Chapelle and Wu, 2010; Chakrabarti *et al.*, 2008), mean reciprocal rank (MRR) (Chakrabarti *et al.*, 2008), and partial AUC (Narasimhan and Agarwal, 2013a). The main challenge in these approaches is the non-convexity of their formulation. One may have to compromise to solve a relaxed convex surrogate. Some methods need to solve quadratic programming with exponential number of constraints and thus are not efficient enough.

The second type of approach (Li *et al.*, 2014b; Agarwal, 2011) seeks to maximize the accuracy at the absolute top of the ranking list. The InfinitePush (Agarwal, 2011), viewed as an extreme case of P-norm Push (Rudin, 2009), uses max-margin principle to construct a scoring function where the largest fraction of *relevant-first-irrelevant-last* violations for each irrelevant instance is minimized. It has shown promising performance in recommendation systems (Christakopoulou and Banerjee, 2005). However, the InfinitePush needs to evaluate all relevant-irrelevant pairs which is computationally expensive. To alleviate the computational burden, the TopPush algorithm (Li *et al.*, 2014b) minimizes the number of relevant instances ranked lower than the topmost irrelevant instance. The TopPush has been shown to learn an equivalent ranking model to the InfinitePush but only needs to evaluate dual variables in linear number of training instances, thus enjoying high efficiency. However, the over-reliance on the topmost irrelevant instance is suspected to be sensitive to outliers and may lead to less robust predictions.

The fragility of the TopPush is due to its excessive emphasis on the single topmost irrelevant instance. To improve the robustness and find a reliable representation of the top-ranked unknown instances, we propose a novel approach, called **SortPush**, which minimizes the fraction of relevant instances ranked lower than a representative of multiple top-ranked irrelevant instances instead of the topmost one. Specifically, SortPush automatically identifies the top-ranked irrelevant instances according to a pre-specified set of weights and uses their weighted combination as the representative of the top irrelevant instances for modeling. For example, one may pre-specify k positive weights so that k irrelevant instances will be selected and linearly combined for modeling. The proposed model includes the TopPush model as a special case when only one of the weights is set to be positive. Also, the proposed model includes the previously proposed "ranking above average" as a special case when all of the

weights are set to be one over the size of unknown instances. The weighted top-k scheme allows us to overcome the all-zero solution in the TopPush and is therefore a non-trivial extension.

The proposed formulation is challenging to solve as the model is based on the ranking order of the irrelevant instances which is also unknown. In this thesis, we adopt the Alternating Direction Method of Multipliers (ADMM) framework (Boyd *et al.*, 2011) where the updating steps involve two optimization subproblems: one is a non-smooth unconstrained problem and the other one is a non-smooth constrained problem. The improved dual updating step is a one-dimensional bisection root finding problem. The dual problem of the non-smooth unconstrained problem can be efficiently solved via accelerated gradient type methods (Beck and Teboulle, 2009; Nesterov, 1983) where the associated proximal operator is shown to admit a closed form solution. As the main technical contribution of this section, we show that the multivariate dual problem of the highly non-trivial non-smooth constrained problem can be converted to a one dimensional concave maximization problem and thus can be efficiently solved by methods such as bisection root finding in logarithmic time.

In the remainder of this section, we first propose the SortPush framework and then derive the optimization procedures. We then report the results of empirical experiments on several ranking benchmark datasets.

3.7 Using Top-ranked unlabeled instances

In dyadic settings, one may construct all feature interactions and rewrite $\tilde{\mathbf{x}}W\tilde{\mathbf{z}}$ as $\mathbf{w}^T\mathbf{x}$ where $\mathbf{x}$ represents all the interactions between $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{z}}$, and $\mathbf{w}$ is the vector reshaped from W. In the following, we consider ranker $\mathbf{w}^T\mathbf{x}$ for clear presentation. Please note that besides the dyadic positive unlabeled learning problems, the proposed method can be applied for arbitrary bipartite ranking problems. We propose to

estimate the scoring function $f(\cdot)$ by minimizing the following ranking loss:

$$\min_f \frac{1}{m}\sum_{i=1}^{m}\ell\left(\sum_{j=1}^{n}\alpha_j f(\mathbf{x}^-_{(j)}) - f(\mathbf{x}^+_i)\right), \tag{3.33}$$

where $-$ represents the irrelevant (unlabeled) class, $+$ represents the relevant (positive) class, $f(\mathbf{x}^-_{(1)}) \geq f(\mathbf{x}^-_{(2)}) \geq \ldots \geq f(\mathbf{x}^-_{(m)})$ are the ordered statistics of scores $f(\mathbf{x}^-)$, and α_j's are a sequence of pre-specified non-negative weights sorted in descending order:

$$\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_n \geq 0, \quad \sum_{j=1}^{n}\alpha_j = 1. \tag{3.34}$$

$\succeq$ denotes the elementwise greater than or equal to operator. Possible loss functions include logistic loss $\log(1+e^x)$, hinge loss $[1+x]_+ \triangleq \max(1+x, 0)$, etc. Intuitively, the formulation in (3.33) computes a representative score linearly combining the ranking scores of unlabeled instances according to the sorted weights α_j's. Then it minimizes the cost of the violations where the representative score is larger than the ranking scores of positive instances.

We name the family of algorithms using the loss (3.33) as **SortPush**. One special case is to form a representative score of top-k ranked unlabeled instances by assigning positive weights to $\alpha_1, \ldots, \alpha_k$ and setting the remaining $n-k$ weights $\alpha_{k+1}, \ldots, \alpha_n$ to zero. For example, when $\alpha_1 = \alpha_2 = \alpha_3 = \frac{1}{3}$, the formulation in (3.33) uses the average score of the top-3 unlabeled instances as the representative of top ranked unlabeled instances for model construction.

In the following, let α_j's be defined as in (3.34). The optimization problem for SortPush can be written as:

$$\min_{\mathbf{w}} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{m}\sum_{i=1}^{m}\ell\left(\sum_{j=1}^{n}\alpha_j \mathbf{w}^T\mathbf{x}^-_{(j)} - \mathbf{w}^T\mathbf{x}^+_i\right), \tag{3.35}$$

where $\mathbf{w}^T\mathbf{x}^-_{(1)} \geq \mathbf{w}^T\mathbf{x}^-_{(2)} \geq \ldots \geq \mathbf{w}^T\mathbf{x}^-_{(m)}$. We first show that Eq.(3.35) is convex when α_j's are in descending order based on the following proposition (Usunier *et al.*, 2009).

Proposition 3.7.1. *Let $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ be a sequence of n non-negative numbers and the sorted weighting function $s_{\boldsymbol{\alpha}} : \mathbb{R}^n \to \mathbb{R}$ be defined as*

$$s_{\boldsymbol{\alpha}}(\mathbf{t}) = \sum_{i=1}^{n} \alpha_j t_{(j)},$$

For $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_n \geq 0$, then

$$s_{\boldsymbol{\alpha}}(\mathbf{t}) = \max_{\sigma \in \Xi} \sum_{j=1}^{n} \alpha_j t_{\sigma(j)},$$

where Ξ is set of permutation mappings, and therefore $s_{\boldsymbol{\alpha}}(\cdot)$ is convex.

The conclusion above is a direct result from the rearrangement inequality. Therefore, $\sum_{j=1}^{n} \alpha_j \mathbf{w}^T \mathbf{x}_{(j)}^-$ is a convex function of $\mathbf{w}$ as it is the composition of a convex function and an affine function of $\mathbf{w}$. Hence, the convexity of problem (3.35) follows from the fact that both ℓ_2 norm and loss $\ell(\cdot)$ are convex (Boyd and Vandenberghe, 2004).

The ADMM Algorithm for Solving SortPush

We consider the hinge loss function, *i.e.,* $\ell(x) = [1 + x]_+$ and propose to use the ADMM to solve the problem (3.35). Denote $X^- \in \mathbb{R}^{n \times d}$ as the data matrix for unlabeled instances where $X_{j,\cdot}^- = \mathbf{x}_j^{-T}$, and $X^+ \in \mathbb{R}^{m \times d}$ as the data matrix for positive instances where $X_{i,\cdot}^+ = \mathbf{x}_i^{+T}$. Let . Let $\mathbf{b} = X^- \mathbf{w} \in \mathbb{R}^n$ and

$$\ell_{\mathcal{H}}(\mathbf{w}, \mathbf{b}) = \frac{1}{m} \sum_{i=1}^{m} \left[1 + \sum_{j=1}^{n} \alpha_j b_{(j)} - \mathbf{w}^T \mathbf{x}_i^+ \right]_+ . \tag{3.36}$$

Then problem (3.35) is equivalent to the following constrained minimization problem:

$$\begin{aligned} \min_{\mathbf{w}, \mathbf{b}} \quad & \frac{\lambda}{2} \|\mathbf{w}\|_2^2 + \ell_{\mathcal{H}}(\mathbf{w}, \mathbf{b}) \triangleq \mathfrak{S}(\mathbf{w}, \mathbf{b}) \\ \text{s.t.} \quad & \mathbf{b} = X^- \mathbf{w} \end{aligned} \tag{3.37}$$

where $b_{(1)} \geq b_{(2)} \geq \ldots \geq b_{(n)}$ are the order statistics of the entries of $\mathbf{b}$, namely, the values of b_i's are sorted in descending order. Define

$$\Delta(\mathbf{w}, \mathbf{b}, \boldsymbol{\xi}) \triangleq \frac{\rho}{2} \left\| X^{-}\mathbf{w} - \mathbf{b} + \boldsymbol{\xi} \right\|_2^2 .$$

The ADMM procedure at iteration t consists of the following three steps:

$$\begin{aligned} \mathbf{w}^{(t+1)} &= \arg\min_{\mathbf{w}} \mathfrak{S}(\mathbf{w}, \mathbf{b}^{(t)}) + \Delta(\mathbf{w}, \mathbf{b}^{(t)}, \boldsymbol{\xi}^{(t)}) \\ \mathbf{b}^{(t+1)} &= \arg\min_{\mathbf{b}} \mathfrak{S}(\mathbf{w}^{(t+1)}, \mathbf{b}) + \Delta(\mathbf{w}^{(t+1)}, \mathbf{b}, \boldsymbol{\xi}^{(t)}) \\ \boldsymbol{\theta}^{(t+1)} &= \boldsymbol{\theta}^{(t)} + X^{-}\mathbf{w}^{(t+1)} - \mathbf{b}^{(t+1)} \end{aligned} \tag{3.38}$$

$\boldsymbol{\theta} \in \mathbb{R}^n, \boldsymbol{\theta} \succeq \mathbf{0}$ is the Lagrangian dual variable and $\rho > 0$ is the tuning parameter. Next, we present the details for solving subproblems associated with $\mathbf{w}$ and $\mathbf{b}$ in (3.38).

Updating w

We consider a simpler form of the optimization problem associated with $\mathbf{w}$ in (3.38) :

$$\min_{\mathbf{w}} \frac{1}{m} \sum_{i=1}^{m} \left[\pi - \mathbf{w}^T \mathbf{x}_i^+\right]_+ + \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \frac{\rho}{2} \left\| X^{-}\mathbf{w} + \boldsymbol{\nu} \right\|_2^2, \tag{3.39}$$

where $\pi := 1 + \sum_{j=1}^{n} \alpha_j b_{(j)}^{(t)}$, and $\boldsymbol{\nu} := -\mathbf{b}^{(t)} + \boldsymbol{\theta}^{(t)}$.

By introducing the slack variables ζ_i's, problem (3.39) can be transformed to the following constrained quadratic programming problem:

$$\begin{aligned} \min_{\mathbf{w},\boldsymbol{\zeta}} \ & \frac{1}{m} \sum_{i=1}^{m} \zeta_i + \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \frac{\rho}{2} \left\| X^{-}\mathbf{w} + \boldsymbol{\nu} \right\|_2^2 \\ \text{s.t.} \ & \left.\begin{array}{l} \zeta_i \geq \pi - \mathbf{w}^T \mathbf{x}_i^+ \\ \zeta_i \geq 0, \end{array}\right\}, \quad i = 1, \ldots, m. \end{aligned} \tag{3.40}$$

From the first-order optimality condition of its Lagrangian function, one can remove the slack variables $\boldsymbol{\zeta}$ and obtain

$$\mathbf{w} = \left(\lambda \mathcal{I} + \rho X^{-T} X^{-}\right)^{-1} \mathbf{m}, \tag{3.41}$$

and the dual problem of (3.40):

$$\begin{aligned} &\min_{\boldsymbol{\delta}} \frac{1}{2}\mathbf{m}^T \left(\lambda\mathbb{I} + \rho X^{-T}X^{-}\right)^{-1}\mathbf{m} - \pi\boldsymbol{\delta}^T\mathbf{1} \\ &\text{s.t. } 0 \leq \delta_i \leq \frac{1}{m}, \quad i = 1, \ldots, m. \end{aligned} \tag{3.42}$$

where $\boldsymbol{\delta}$ is the dual variable, $\delta_i \geq 0,\ i = 1, \ldots, n,\ \mathbf{m} = \left(X^{+T}\boldsymbol{\delta} - \rho X^{-T}\boldsymbol{\nu}\right)$ and $\mathbb{I} \in \mathbb{R}^{n\times n}$ is the identity matrix. It can be easily verified that the proximal operator associated with problem (3.42) admits a closed form solution. Therefore, after we obtain a dual optimal solution $\boldsymbol{\delta}^*$ that maximizes the problem (3.42), we can readily compute the closed form proximal operator $\mathbf{w}^*$ via the closed form solution of $\mathbf{w}$ in (3.41) and problem (3.42) can thus be efficiently solved via accelerated gradient algorithms (Beck and Teboulle, 2009; Nesterov, 1983). Note that the matrix inversion only needs to be computed once through out the algorithm.

3.7.1 Updating $\mathbf{b}$

We simplify the notations of the optimization problem associated with $\mathbf{b}$ and have:

$$\min_{\mathbf{b}} \frac{1}{m}\sum_{i=1}^{m}\left[\sum_{j=1}^{n}\alpha_j b_{(j)} - u_i\right]_+ + \frac{\rho}{2}\left\|\mathbf{b} - \mathbf{v}\right\|_2^2, \tag{3.43}$$

where $\mathbf{u} := -\mathbf{1} + X^{+}\mathbf{w}^{(t+1)}$, and $\mathbf{v} := X^{-}\mathbf{w}^{t+1} + \boldsymbol{\theta}^{(t)}$. Since the ℓ_2 norm is invariant of permutation, we make the following assumption without loss of generality:

Assumption 1. *The vector* $\mathbf{v} \in \mathbb{R}^n$ *obeys* $v_1 \geq v_2 \geq \ldots \geq v_n$.

Note that one can always obtain the solution of $\mathbf{b}$ by applying the inverse of the permutation that sorts $\mathbf{v}$ in descending order.

Proposition 3.7.2. *Under Assumption 1, the solution* $\mathbf{b}$ *to problem* (3.43) *satisfies* $b_1 \geq b_2 \geq \ldots \geq b_n$.

Proof. for Proposition 3.7.2 The proof is analogous to the proof of Proposition 2.2 stated in (Bogdan *et al.*, 2013).

Suppose that $b_i < b_j$ for $i < j$ (and $v_i > v_j$), and form a copy $\mathbf{b}'$ of $\mathbf{b}$ with entries i and j exchanged. Letting $\mathcal{L}$ be the objective functional in (3.43), we have

$$f(\mathbf{b}) - f(\mathbf{b}') = \frac{1}{2}(v_i - b_i)^2 + \frac{1}{2}(v_j - b_j)^2 - \frac{1}{2}(v_i - b_j)^2 - \frac{1}{2}(v_j - b_i)^2. \tag{3.44}$$

This follows from the fact that the sorted linear combination in Part I takes on the same value at $\mathbf{b}$ and $\mathbf{b}'$ and that all the quadratic terms cancel but those for i and j. This gives

$$f(\mathbf{b}) - f(\mathbf{b}') = (b_i - b_j)(v_j - v_i) > 0, \tag{3.45}$$

which shows that the objective of $\mathbf{b}'$ is strictly smaller, which contradicts to the optimality of $\mathbf{b}$. □

Therefore, problem (3.43) can be transformed to the constrained problem:

$$\begin{aligned} &\min_{\mathbf{b}} \frac{1}{m}\sum_{i=1}^{m}\left[\sum_{j=1}^{n}\alpha_j b_j - u_i\right]_+ + \frac{\rho}{2}\left\|\mathbf{b}-\mathbf{v}\right\|_2^2 \\ &\text{s.t. } b_1 \geq b_2 \geq \ldots \geq b_n. \end{aligned} \tag{3.46}$$

As the first term in (3.46) is non-smooth, we introduce the slack variables ξ_i's and transform problem (3.46) to the constrained quadratic programming problem:

$$\begin{aligned} &\min_{\mathbf{b},\boldsymbol{\xi}} \frac{1}{m}\sum_{i=1}^{m}\xi_i + \frac{\rho}{2}\|\mathbf{b}-\mathbf{v}\|_2^2 \\ &\text{s.t. } \left.\begin{array}{ll} b_1 \geq b_2 \geq \ldots \geq b_n, & \\ \xi_i \geq \sum_{j=1}^{n}\alpha_j b_j - u_i & \\ \xi_i \geq 0, & \end{array}\right\}, \quad i = 1,\ldots,m \end{aligned} \tag{3.47}$$

The Lagrangian function associated with the problem (3.47) is given by

$$\mathfrak{L}(\mathbf{b},\boldsymbol{\xi},\boldsymbol{\gamma},\boldsymbol{\eta}) = \frac{1}{m}\sum_{i=1}^{m}\xi_i + \frac{\rho}{2}\|\mathbf{b}-\mathbf{v}\|_2^2 - \sum_{i=1}^{m}\gamma_i\left(\xi_i - \sum_{j=1}^{n}\alpha_j b_j + u_i\right) - \sum_{i=1}^{m}\eta_i\xi_i \tag{3.48}$$

where $\gamma_i \geq 0, \eta_i \geq 0, i = 1, \ldots, n$ are Lagrangian dual variables. The first-order optimality condition $\frac{\partial \ell}{\partial \xi_i} = 0$ results in

$$0 \leq \gamma_i \leq \frac{1}{m}, \quad i = 1, \ldots, n. \tag{3.49}$$

Then, the Lagrangian function (3.48) can be simplified as

$$\ell(\mathbf{b}, \boldsymbol{\gamma}) = \frac{\rho}{2}\|\mathbf{b} - \mathbf{v}\|_2^2 + \sum_{i=1}^{m} \gamma_i \left(\sum_{j=1}^{n} \alpha_j b_j - u_i \right). \tag{3.50}$$

Thus, the dual problem of (3.47) becomes

$$\begin{aligned} \max_{\boldsymbol{\gamma}} \min_{\mathbf{b}} \; & \frac{\rho}{2}\|\mathbf{b} - \mathbf{v}\|_2^2 + \sum_{i=1}^{m} \gamma_i \left(\sum_{j=1}^{n} \alpha_j b_j - u_i \right) \\ \text{s.t. } & b_1 \geq b_2 \geq \ldots \geq b_n, \\ & 0 \leq \gamma_i \leq \frac{1}{m}, \quad i = 1, \ldots, n \end{aligned} \tag{3.51}$$

We next show that problem (3.51) can be transformed to an equivalent one dimensional optimization problem.

First, we write the objective of minimization problem (3.51) as a function of $c = \sum_{i=1}^{n} \gamma_i$:

$$\begin{aligned} g_1(c) := \min_{\mathbf{b}} \; & \frac{\rho}{2}\|\mathbf{b} - \mathbf{v}\|_2^2 + c \sum_{j=1}^{n} \alpha_j b_j \\ \text{s.t. } & b_1 \geq b_2 \geq \ldots \geq b_n \end{aligned}. \tag{3.52}$$

Then, problem (3.51) is equivalent to the following maximization problem w.r.t $\boldsymbol{\gamma}$ and c:

$$\begin{aligned} \max_{\boldsymbol{\gamma}, c} \; & g_1(c) - \sum_{i=1}^{m} \gamma_i u_i \\ \text{s.t. } & c = \sum_{i=1}^{m} \gamma_i, \quad 0 \leq \gamma_i \leq \frac{1}{m}, \quad i = 1, \ldots, m \end{aligned}. \tag{3.53}$$

Furthermore, the affine function of γ_i's can be written as another function of c, *i.e.,*

$$\begin{aligned} g_2(c) := \min_{\boldsymbol{\gamma}} \ & \sum_{i=1}^{m} \gamma_i u_i \\ \text{s.t.} \ & \sum_{i=1}^{m} \gamma_i = c, \\ & 0 \leq \gamma_i \leq \frac{1}{m}, \quad i = 1, \ldots, m \end{aligned} \qquad . \tag{3.54}$$

Therefore, problem (3.51) can be rewritten as the following:

$$\begin{aligned} \max_{c} \mathcal{G}(c) = \max_{c} \ & g_1(c) - g_2(c) \\ \text{s.t.} \ & 0 \leq c \leq 1 \end{aligned} \tag{3.55}$$

To this end, we transform the multivariate optimization problem (3.51) to a one dimensional optimization problem. We summarize the properties of problem (3.55) in the following theorem.

Theorem 3.7.3. *Problem* (3.55) *is a one dimensional concave maximization problem and*

- $g_1(c)$ *is a concave function and has a continuous derivative;*
- $g_2(c)$ *is a piecewise linear function of* c *and is a convex function.*

Proof. for Theorem 3.7.3 First of all, since function $g_1(c)$ is a pointwise infimum of affine functions of c, it follows that $g_1(c)$ is concave (Boyd and Vandenberghe, 2004). Based on similar analysis of Lemma 2 in (Chapelle *et al.*, 2002), given $c = c_0$, one can compute the the derivative of $g_1(c)$, *i.e.,*

$$\left.\frac{\partial g_1(c)}{\partial c}\right|_{c=c_0} = \sum_{j=1}^{n} \alpha_j b_j^*, \tag{3.56}$$

where $\mathbf{b}^*$ is the optimal solution to problem (3.52) at $c = c_0$. For a fixed c, the exact solution to problem (3.52) can be computed in $\mathcal{O}(n)$ time which is obtained in a way

analogous to the algorithm for the proximal operator problem stated in (Bogdan *et al.*, 2013). As $g_1(c)$ is concave and differentiable w.r.t c everywhere in the range $[0, 1]$, its derivative $\frac{\partial g_1(c)}{\partial c}$ is continuous according to Theorem 25.5 in (Rockafellar, 1970). We next prove that function $g(c)$ defined in (3.54) is a piecewise linear function. We first assume u_i's have already been sorted in descending order without loss of generality, *i.e.,*

$$u_1 \leq u_2 \leq \ldots \leq u_m.$$

And we denote $q = \lfloor cm \rfloor$ and $r = c - \frac{\lfloor cm \rfloor}{m}$. Then, for a given c, the solution to (3.54) would be $\gamma_1 = \frac{1}{m}, \gamma_2 = \frac{1}{m}, \ldots, \gamma_q = \frac{1}{m}, \gamma_{q+1} = r$ and $\gamma_i = 0$ for $i = q+2, \ldots, m$. Thus, it is obvious that for $c \in [0, 1]$, the function value $g(c)$ is a piecewise linear function in intervals $\left[\frac{i-1}{m}, \frac{i}{m}\right]$ with slopes u_i, $i = 1, \ldots, m$. Moreover, let $z_i = \frac{1}{m}\sum_{j=1}^{i-1} u_j - \frac{i-1}{m}u_i, i = 1, \ldots, m$, then $g_2(c)$ can be expressed as

$$g_2(c) = \max_{i=1,\ldots,m} u_i c + z_i. \tag{3.57}$$

Hence, $g_2(c)$ is convex (Boyd and Vandenberghe, 2004) and $-g_2(c)$ is thus concave. Since the constraint function in (3.55) is linear and the objective is a linear combination of two concave functions, problem (3.55) is a one dimensional concave maximization problem, which completes the proof. □

Based on the result above, we convert problem (3.55) to a root finding problem and propose a modified bisection algorithm to solve it. For intervals $\left(\frac{i-1}{m}, \frac{i}{m}\right), i = 1, \ldots, m$, the derivative of $\mathcal{F}(c)$ is

$$\frac{\partial \mathcal{G}(c)}{\partial c} = \frac{\partial g_1(c)}{\partial c} - \frac{\partial g_2(c)}{\partial c}, \tag{3.58}$$

where $\frac{\partial g_1(c)}{\partial c}$ is computed via (3.56) and

$$\frac{\partial g_2(c)}{\partial c} = u_i. \tag{3.59}$$

For the breakpoints $c \in \left\{\frac{i}{m} : i = 1, \ldots, m-1\right\}$ where $g_2(c)$ is non-differentiable, we know the corresponding subdifferential satisfies

$$\frac{\partial g_2(c)}{\partial c} \in [u_i, u_{i+1}]. \tag{3.60}$$

And for the boundary points 0 and 1, the subdifferential is in $(-\infty, u_1]$ and $[u_m, \infty)$ seperately. Therefore, $\frac{\partial \mathcal{G}(c)}{\partial c}$ is guaranteed to be monotonically decreasing. The detailed algorithm for solving problem (3.55) is summarized in Algorithm 3. The details of computing $\frac{\partial \mathcal{G}(c)}{\partial c}$ are also elaborated in the supplement. Once c is found by Algorithm 3, we will obtain the optimal solution to problem (3.43), *i.e.,* $\mathbf{b}^*$ that solves problem (3.52).

3.7.2 The Choice of Weighting Scheme

The weighting coefficients α_i controls the loss penalty when a mistake is made in the top-k instances. Here we focus on the top-k polynomial weighting scheme:

$$\alpha_i = \begin{cases} i^{-p}/(\sum_{j=1}^{k} j^{-p}) & 1 \leq i \leq k \\ 0 & i > k \end{cases}.$$

When $p \to \infty$, SortPush reverts to TopPush; when $p = 0$, SortPush averages the top-k negative instances. It is possible to set $p < 0$. In this case, SortPush will tolerate mistake on the topmost negative instance, and try to optimize the k-th negative instance in the top of the ranking list. This allows SortPush to be more robust to noise, especially when the data is not linearly separable. By setting $p < 0$, SortPush will pay more attention to negative instances that are beyond the noise margin and ignore instances within the noise margin.

Algorithm 3: Computing c in (3.55)

Input: scalar ρ, vector $\mathbf{u}$, $\mathbf{v}$ and $\boldsymbol{\alpha}$ sorted in descending order

Output: $\bar{c}$

$\frac{\partial \mathcal{G}(c)}{\partial c} = \frac{\partial g_1(c)}{\partial c} - \frac{\partial g_2(c)}{\partial c}$ at $c_1 = 0$, $c_2 = 1$

if $\frac{\partial \mathcal{G}(c)}{\partial c}\big|_{c=c_1}$ and $\frac{\partial \mathcal{G}(c)}{\partial c}\big|_{c=c_2}$ have the same sign **then**

  Choose $\bar{c} = \arg\max_{c \in \{c_1, c_2\}} \mathcal{G}(c)$

else

  $\bar{c} = \frac{c_1 + c_2}{2}$;

  while $0 \notin \frac{\partial \mathcal{G}(c)}{\partial c}\big|_{c=\bar{c}}$ **do**

    Compute $\frac{\partial \mathcal{G}(c)}{\partial c}$ at $\bar{c} = \frac{c_1 + c_2}{2}$

    if $\frac{\partial \mathcal{G}(c)}{\partial c}\big|_{c=\bar{c}}$ has different sign from $\frac{\partial \mathcal{G}(c)}{\partial c}\big|_{c=c_1}$ **then**

      $c_2 = \bar{c}$; $\frac{\partial \mathcal{G}(c)}{\partial c}\big|_{c=c_2} = \frac{\partial \mathcal{G}(c)}{\partial c}\big|_{c=\bar{c}}$;

    else

      $c_1 = \bar{c}$; $\frac{\partial \mathcal{G}(c)}{\partial c}\big|_{c=c_1} = \frac{\partial \mathcal{G}(c)}{\partial c}\big|_{c=\bar{c}}$;

    end if

  end while

end if

3.8 Experiments on comparing SortPush with other bipartite ranking methods

For more comprehensive comparisons, we compare the proposed SortPush model with the popular baseline methods for bipartite ranking on four well-known benchmark datasets for learning to rank.

3.8.1 Data Description

The benchmark datasets used in our study are the TREC 2003 and TREC 2004 (called TD2003 and TD2004 from LETOR 2.0) dataset, and the TREC 2007 and TREC 2008 (called MQ2007 and MQ2008 from LETOR 4.0) dataset which are pub-

Table 3.1: Dataset Statistics: d is the dimension, m is the number of positive instances, n is the number of negative instances

Dataset	d	$n+m$	m	n
TD2003	44	49171	516	48655
TD2004	44	74170	444	73726
MQ2007	41	69623	3863	65760
MQ2008	40	15211	931	14280

licly available for download from Microsoft Research [5] . The relevance scores in TD2003 and TD2004 datasets are from $\{0, 1\}$ while those in MQ2007 and MQ2008 range from $\{0, 1, 2\}$. As we examine the performance for the bipartite ranking, we group items with relevance score 2 into the set of relevant instances in the experiment. All the features are normalized using z-score where each feature is subtracted by its mean and divided by its standard deviation. Detailed data statistics are shown in Table 3.1.

3.8.2 Experiment Settings

We compare the proposed SortPush with six baseline methods: **TopPush** (Li *et al.*, 2014b) and Support Vector Machine (**SVM**), **SVMpAUC** (Narasimhan and Agarwal, 2013b), RankNet (Burges *et al.*, 2005) that optimizing mAP (**RN-mAP**) / NDCG (**RN-NDCG**) / Precision (**RN-Prec**). The implementation of TopPush [6] and SVMpAUC [7] are released by the authors. We use liblinear to implement SVM

[5]http://research.microsoft.com/en-us/um/beijing/projects/ letor/

[6]http://lamda.nju.edu.cn/code TopPush.ashx

[7]http://clweb.csa.iisc.ernet.in/harikrishna/Papers/SVMpAUC/

[8] . The SVMpAUC is released by the author The algorithm RankeNet used is from software package RankLib [9] .

We evaluate the ranking performance using a variety of metrics including the mean Average Precision (mAP), NDCG and AUC score. To examine the results on top of the ranking list, we also compute these scores restricted to the top τ instances, denoted as 'mAP@τ' or 'NDCG@τ'. All datasets are randomly split into 60% training set and 40% testing set. All experiments are repeated 5 times and the mean and standard deviation of the above metrics are reported. For SVM, we tune the parameter C from the set $\{1, 10, 10^2, 10^3, 10^4, 10^5\}$. For the setting of SVMpAUC, the parameter C is chosen from the set $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}$. The number of layers in RankNet is tuned from 1 to 5, and the metric NDCG and Precision are optimized at $\tau = 10$. For SortPush, we choose the top-k parameter from top 1% to top 100% with 20 equally spaced values. Parameter p in the top-k polynomial weighting scheme is tuned from the set $\{0, \pm 0.5, \pm 1, \pm 2, \pm 4\}$. The parameter λ in TopPush and SortPush are tuned the same as SVM with $\lambda = 1/C$. Due to the slow convergence rate of TopPush, we set the maximum number of iteration in TopPush to be 5×10^5. All parameters are selected via cross-validation based on the measurement of mean average precision.

3.8.3 Ranking Performance

We report the ranking performance of SortPush in Table 3.2 and Table 3.3. Methods such as SortPush, SVMpAUC, RankNet generally perform better than SVM, indicating the necessity of optimizing a specific metric for top ranking. Moreover, in most cases, the performances of SortPush are significantly better than those of other

[8]http://www.csie.ntu.edu.tw/ cjlin/liblinear/

[9]https://people.cs.umass.edu/ vdang/ranklib.html

Table 3.2: Performances achieved on TD2003 and TD2004 datasets by SortPush and the baseline approaches.

TD2003	mAP	mAP@5	mAP@10	NDGG	NDGG@5	NDGG@10	AUC
SortPush	$\mathbf{23.5 \pm 5.7}$	$\mathbf{23.2 \pm 4.2}$	$\mathbf{20.2 \pm 4.8}$	$\mathbf{49.4 \pm 3.3}$	$\mathbf{64.1 \pm 3.9}$	$\mathbf{62.5 \pm 4.9}$	$\mathbf{80.3 \pm 4.7}$
SVM	11.8 ± 11.2	11.9 ± 9.3	10.1 ± 8.8	34.7 ± 8.0	39.4 ± 11.3	38.6 ± 18.5	59.7 ± 18.0
TopPush	9.1 ± 20.2	8.5 ± 5.5	7.2 ± 5.6	32.0 ± 4.6	25.2 ± 10.3	26.4 ± 14.8	55.2 ± 16.0
SVMpAUC	18.7 ± 6.7	18.4 ± 4.7	15.3 ± 5.7	45.1 ± 4.4	56.9 ± 7.3	57.4 ± 17.4	74.6 ± 12.7
RN-mAP	11.3 ± 10.7	11.3 ± 3.3	9.4 ± 3.8	35.9 ± 2.6	35.3 ± 4.8	36.4 ± 10.3	69.7 ± 9.6
RN-NDCG	11.6 ± 9.6	11.0 ± 3.4	9.3 ± 3.9	36.4 ± 2.6	38.2 ± 3.3	40.0 ± 10.7	68.7 ± 9.8
RN-Prec	16.4 ± 6.1	12.6 ± 3.4	12.5 ± 2.9	42.5 ± 3.8	50.0 ± 3.5	49.1 ± 7.5	74.2 ± 7.8
TD2004	mAP	mAP@5	mAP@10	NDGG	NDGG@5	NDGG@10	AUC
SortPush	$\mathbf{35.8 \pm 0.6}$	$\mathbf{31.8 \pm 5.4}$	$\mathbf{31.3 \pm 6.5}$	$\mathbf{58.7 \pm 5.6}$	$\mathbf{66.9 \pm 5.5}$	$\mathbf{66.5 \pm 7.0}$	$\mathbf{94.8 \pm 3.3}$
SVM	11.1 ± 7.2	8.5 ± 2.4	8.2 ± 4.3	32.4 ± 5.3	24.3 ± 2.2	25.5 ± 8.7	74.5 ± 8.2
TopPush	11.3 ± 9.8	7.8 ± 4.8	8.3 ± 3.1	33.4 ± 3.7	27.9 ± 5.9	29.9 ± 9.7	67.7 ± 6.0
SVMpAUC	33.1 ± 0.4	27.5 ± 5.2	28.3 ± 6.6	55.5 ± 5.1	62.0 ± 4.2	62.1 ± 5.8	94.4 ± 4.5
RN-mAP	25.9 ± 2.0	22.4 ± 6.0	21.6 ± 8.1	47.9 ± 6.0	49.6 ± 6.1	52.0 ± 9.4	88.5 ± 6.9
RN-NDCG	23.8 ± 4.8	19.6 ± 6.8	19.6 ± 4.5	45.2 ± 5.4	48.1 ± 7.5	47.1 ± 9.5	86.9 ± 10.1
RN-Prec	22.6 ± 5.6	18.3 ± 7.5	18.5 ± 8.3	42.8 ± 7.5	45.1 ± 13.2	43.2 ± 20.0	85.8 ± 18.6

Table 3.3: Performances achieved on MQ2007 and MQ2008 datasets by SortPush and the baseline approaches.

MQ2007	mAP	mAP@5	mAP@10	NDGG	NDGG@5	NDGG@10	AUC
SortPush	$\mathbf{49.0 \pm 0.8}$	$\mathbf{39.7 \pm 1.9}$	$\mathbf{42.2 \pm 2.0}$	$\mathbf{64.9 \pm 1.7}$	$\mathbf{63.0 \pm 1.7}$	$\mathbf{63.5 \pm 1.8}$	$\mathbf{70.2 \pm 2.2}$
SVM	45.5 ± 2.0	35.6 ± 1.7	38.0 ± 2.3	62.0 ± 1.8	58.5 ± 1.6	60.7 ± 2.0	67.7 ± 1.6
TopPush	33.5 ± 11.0	22.2 ± 7.8	24.6 ± 8.2	51.7 ± 8.7	44.5 ± 7.4	46.8 ± 12.4	56.5 ± 11.1
SVMpAUC	36.1 ± 5.3	25.0 ± 3.5	27.6 ± 3.9	54.3 ± 2.8	48.8 ± 3.3	51.5 ± 2.6	61.8 ± 2.6
RN-mAP	46.8 ± 1.5	37.3 ± 1.0	39.6 ± 1.2	62.8 ± 0.9	59.8 ± 1.2	61.1 ± 1.3	67.5 ± 1.1
RN-NDCG	46.6 ± 1.5	37.0 ± 1.0	39.4 ± 1.1	62.8 ± 0.9	59.6 ± 1.0	61.1 ± 1.1	67.4 ± 1.1
RN-Prec	46.5 ± 1.6	36.8 ± 1.0	39.3 ± 1.2	62.7 ± 1.2	59.6 ± 1.0	61.0 ± 1.1	67.4 ± 1.1
MQ2008	mAP	mAP@5	mAP@10	NDGG	NDGG@5	NDGG@10	AUC
SortPush	$\mathbf{64.4 \pm 2.0}$	$\mathbf{59.8 \pm 4.5}$	$\mathbf{62.5 \pm 4.8}$	$\mathbf{78.4 \pm 5.0}$	$\mathbf{79.0 \pm 3.3}$	$\mathbf{79.1 \pm 2.7}$	$\mathbf{80.4 \pm 3.2}$
SVM	59.5 ± 2.3	53.4 ± 3.1	57.3 ± 3.3	74.1 ± 3.2	72.7 ± 2.2	74.1 ± 3.5	78.0 ± 2.3
TopPush	39.1 ± 11.2	30.7 ± 8.9	35.4 ± 10.1	57.4 ± 9.6	51.1 ± 7.7	55.3 ± 9.5	52.1 ± 9.0
SVMpAUC	55.1 ± 3.8	48.7 ± 2.3	52.5 ± 2.6	70.7 ± 2.5	69.7 ± 2.0	70.8 ± 3.0	71.7 ± 2.5
RN-mAP	60.9 ± 3.1	55.0 ± 4.2	58.7 ± 4.8	75.0 ± 4.6	73.7 ± 4.1	75.0 ± 5.2	78.4 ± 4.5
RN-NDCG	60.9 ± 2.9	55.2 ± 4.7	58.6 ± 5.2	75.0 ± 5.1	73.7 ± 3.7	75.0 ± 5.3	78.4 ± 4.0
RN-Prec	61.2 ± 3.0	55.5 ± 4.9	59.0 ± 5.5	75.4 ± 5.3	73.8 ± 3.8	75.5 ± 4.8	78.6 ± 4.1

baseline method.The poor performance achieved by TopPush shows its fragility to datasets which may be contaminated by outliers/noises. We discover that, in our experiment, TopPush typically converges to all-zero solution for the real datasets, which indicates the vulnerability to outliers. Also, we observe that the variance of the performance of SortPush is usually smaller than the baseline approaches, which indicates the stability of the SortPush. For example, on TD2004, the variance of mAP of SortPush is only 0.6%, while SVM is 7.2% and TopPush is 9.8%.

3.8.4 SortPush Performance under Different Parameters

In order to understand how the parameters affect the ranking performance of SortPush, we plot the performance curves obtained by SortPush under different parameter settings in Figure 3.4. The plots the mean average precision of SortPush where the x-axis is the top-k parameter. We use different line colors to indicate different p in the polynomial weighting scheme for SortPush. In the figure, we highlight the performance of SVM and TopPush for reference. In Figure 3.4, when $p = 4$, SortPush will revert to TopPush and we observe that the performance drops dramatically as expected. To achieve the best performance, one needs a small p and a sufficiently large k. On the MQ2008 dataset, we observe that the highest prediction performances concentrate around $k \geq 80\%$ indicating that the majority of the instances is needed to learn a good ranker. Interestingly, SortPush usually achieves the best performance at $p = 0.5$. One possible reason is that it considers top instances and non-top instances with smoothly changing weights and thus fully takes advantage of the information of the data.

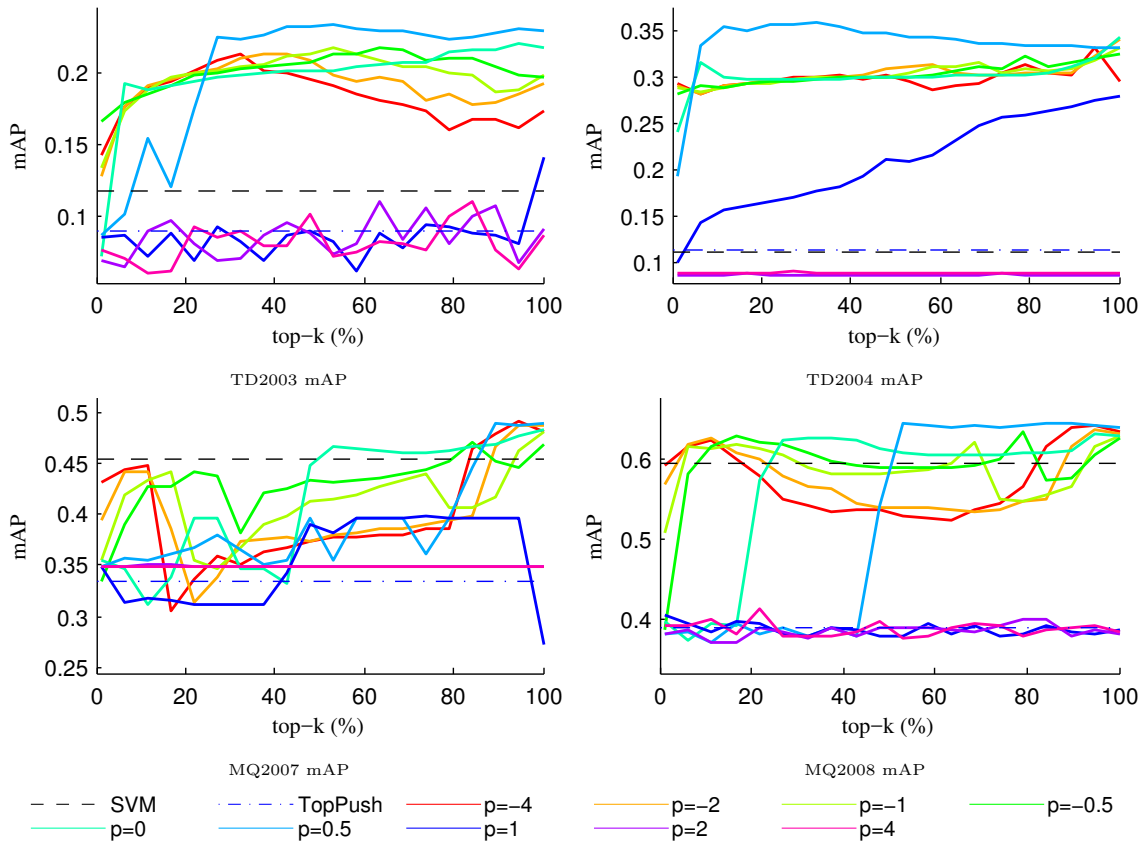

Figure 3.4: The mean average precision curves achieved by the SortPush with vairous top-k% and p in polynomial weighting scheme.

Chapter 4

LEARNING WITH SUBGROUPING

4.1 Introduction

Nowadays, data has been growing larger and more heterogeneous in many areas such as social media, finance, healthcare, agriculture, transportation. These big and heterogeneous data brings new challenges for biulding powerful predictive models with conventional methods. One straightforward solution to this challenge is to divide the data into multiple subgroups that show more homogeneity and build models individually for each subgroup. The development and research of this idea has been recognized for a long time. A simple regression model with a 0-1 dummy variable interacted with other independent variables can be viewed as a direct applicaiton. Consider a regression model,

$$y = w_0 + \sum_{i=1}^{d} w_i x_i + \epsilon, \tag{4.1}$$

where y is a scalar label, $w_0, w_1, \ldots, w_d$ are the unknown regression coefficients and $x_0, x_1, \ldots, \mathbf{x}_d$ correspond to the features, ϵ is the error term. Let $x_I \in \{0, 1\}$ be an indicator variable (e.g., indicator of gender). By including x_I and its interactions with other variables in the regression model, one may obtain

$$y = w_0 + \sum_{i=1}^{d} w_i x_i + u_0 x_I + \sum_{i=1}^{d} u_i x_i x_I + \epsilon, \tag{4.2}$$

where $u_0, u_1, \ldots, u_d$ are the unknown regression coefficients for x_I and the interaction terms. As one can observe, for samples with $x_I = 0$, the model turns into

$$y = w_0 + \sum_{i=1}^{d} w_i x_i + \epsilon. \tag{4.3}$$

For samples with $x_I = 1$, the model turns into

$$y = w_0 + u_0 + \sum_{i=1}^{d}(w_i + u_i)x_i + \epsilon. \tag{4.4}$$

This observation implies that the regression model using interactions composed by an indicator variable essentially learns two models at the same time: one is for the group of $x_I = 0$ where the intercept and slope coefficeints are $w_0, w_1, \ldots, w_d$ and the other one is for the group of $x_I = 1$ where the intercept and slope coefficeints are $w_0+u_0, w_1+u_d, \ldots, w_d+u_d$. If the indicator variables and its interactions are of great importance, it implies that this prior division of data according to the indictor may lead to two seperate groups which are dissimilar but show more homogeneity within the group and thus fitting them with seperate models would be preferred. There are many real-world applications where subgrouping samples play an importance role on data analysis or modeling. For example, it is well known that many diseases show very different patterns in terms of gender.

However, determining a good data division using interactions composed by indicator variables requires strong prior knowledge. Moroever, the number of factors resulting in the data division can be large and thus using interaction models needs testing all the possible combinations which is disastrous for the computation. Clusterwise regression was originally proposed to simultaneously clustering (grouping) and regression (Späth, 1979, 1982, 2014). In clusterwise regression, the size of cluster K is predefined and it solves the following optimization problem:

$$\min_{f_k;\mathcal{C}_k} \sum_{k=1}^{K} \sum_{(\mathbf{x}_i,y_i)\in\mathcal{C}_k} \ell(f_k(\mathbf{x}_i), y_i) \tag{4.5}$$

where $\ell(\cdot)$ is the loss function, $\mathcal{C}_k \cap \mathcal{C}_{k'} = \emptyset$, $\cup_{k=1}^{K}\mathcal{C}_k$ is the whole dataset, $f_k(\cdot)$ corresponds to a regression model for the k-th cluster. A typical algorithm for solving this objective consists of the following procedures: first, randomly initialize K clusters; second, build models for each cluster and shift data points to the cluster that

achieves minimum risidual error; repeat the second procedure until convergence. The original version of clusterwise regression is based on K-Means (KM) which therefore inherited the disadvantages of KM such as sensitive to initializations. Driven by a closing motivation, Deodhar and Ghosh (2007) proposed a framework named Simultaneous Co-Clustering and Learning (SCOAL) for customer-product recommendation where the clustering method is applied on both customer and product dimensions. Zhang (2003) proposed to improve the stability by using harmonic means. DeSarbo and Cron (1988) introduced conditional mixtures to clusterwise regression problem and adopted EM algorithms for parameter estimations which recieved great attention (Hennig, 2000b,a, 1999). Muruzábal *et al.* (2012) proposed a nueral network method for clusterwise regression. As the KM-type clusterwise regression requires residuals for clustering, it is difficult to predict a new instance of which the groundtruth label is unknown. While the algorithms adopting mixture models need extra assumptions and therefore more parameters to estimate, which is not favorable in practice. Moreover, the underlying clustering (grouping) principles of the classic clusterwise regression algorithms are usually not interpretable.

Motivated by the properties of interaction models, we propose a framework in this thesis that enables simultaneously subgrouping data points and building learning models for each separate subgroup. There are two basic assumptions for the proposed framework: (1) subgrouping of the data set is determined by a small portion of features; (2) all models built on the subgrouped data share the same model assumption (e.g., linearity in regression). Compared to the classic clusterwise regression, the proposed framework predicts an unseen data point by adopting a model-based method for the group assignment procedure. Furthermore, the proposed framework is able to identify a group of variables that are critical to the subgrouping procedure. More importantly, our proposed framework has the ability to offer a mechanism to fully utilize

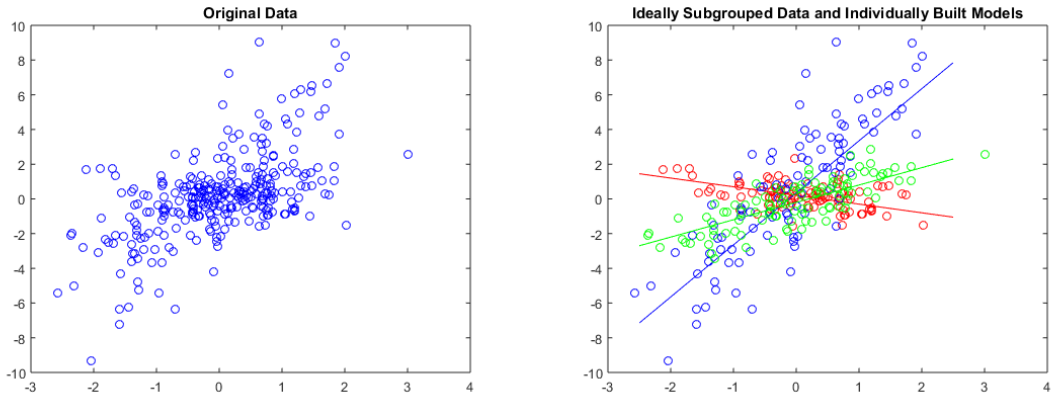

Figure 4.1: Motivation illustration in the regression setting. The top figure: the visualization of all the original data points; The bottom figure: by subgrouping all the data points into three subgroups and fitting regression model separately will gives one satisfactory prediction results.

the information provided across subgroups and not treat subgroups independently. Recently, multi-task learning (Caruana, 1997) has been recieved increasing attention which aims to improve the generalization performance by learning multiple tasks simultaneously and exploiting the intrinsic relations among the tasks. Our proposed framework can easily embed various multi-task learning techniques and thus it can extract relatedness or common knowledge among seperate/independent subgroups.

In the remainder of this section, we first propose the framework for simultaneous subgrouping and learning and then introduce the optimization procedues. We report the experimental results at the end.

4.2 Simultaneous Subgrouping and Learning

4.2.1 The proposed framework

In this section, all the matrices are represented by uppercase letters (e.g., A), vectors are represented by boldface lowercase letters (e.g., $\mathbf{a}$), and entries of a vector or matrix is represented by regular lowercase letters (e.g., a_i, $A_{i,j}$). In the following, we consider the setting of regression. The proposed method and conclusions can be easily extended to the classification setting. We denote y as the scalar outcome and $\mathbf{x} = (x_1, x_2, \ldots, x_d)^T \in \mathbb{R}^d$ as the d-dimensional column feature vector. Suppose we are given n data pairs $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\} \subset \mathbb{R}^d \times \mathbb{R}$ and assume that there are K underlying subgroups $\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_K$. Then, a regression model for the subgroup $\mathcal{G}_k$ is as follows:

$$y = \mathbf{x}^T \mathbf{u}_k + \epsilon, \tag{4.6}$$

where ϵ is the noise term following normal distribution, $\mathbf{u}_k \in \mathbb{R}^d$ is the coefficient vector for the k-th subgroup. Note that outcome y and feature vector $\mathbf{x}$'s are assumed to be centered and thus the bias term is omitted. Let $\mathbf{g} \in \mathbb{R}^K$ be the subgroup indicator variable where the k-th entry is 1 if the data point belongs to subgroup $\mathcal{G}_k$ and 0 otherwise. Let $U = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_K] \in \mathbb{R}^{d \times K}$ and we can represent all models for different subgroups in the following equation:

$$y = \mathbf{x}^T U \mathbf{g} + \epsilon. \tag{4.7}$$

If one knows exactly which subgroup a data point is assigned, one can learn the above model via techniques developed from the multi-task learning framework which has already been well studied. Unfortunately, the pattern of subgroups is typically unknown in many applications. To our best knowledge, there has not been an approach that simultaneously learns the pattern of subgrouping and builds learning

models individually on each subgroup. We next propose to model this problem in one framework. Assume that the subgroup label $\mathbf{g}$ can be represented by the linear combinations of the features, that is:

$$\mathbf{g} \approx V^T \mathbf{x}, \tag{4.8}$$

where $V = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_K] \in \mathbb{R}^{d \times K}$ is the coefficient matrix for subgrouping. One can also model the subgrouping via sigmoid function, i.e.,

$$\mathbf{g} \approx \frac{1}{1 + \exp\left(-V^T \mathbf{x}\right)}, \tag{4.9}$$

which can be solved in a similar way. As we mentioned before, subgrouping is typically determined by a small amount of features and thus can we assume that coefficient matrix V is row-wise sparse, *i.e.,* there are rows of V being all zeros. Let $G = [\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_n]^T \in \mathbb{R}^{n \times K}$ be the subgroup indicator matrix for all the n data points. Then, the loss of learning regression models would be:

$$\ell^r(U, G) = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \mathbf{x}_i^T U \mathbf{g}_i\right)^2. \tag{4.10}$$

The loss of subgrouping would be:

$$\ell^s(G, V) = \frac{1}{n} \sum_{i=1}^{n} \left(\mathbf{g}_i - V^T \mathbf{x}_i\right)^2. \tag{4.11}$$

To control the model complexity, we apply the frobenius norm on the model coefficient matrix U and we use the group lasso penalty on the subgrouping coefficient matrix V for feature selection. Thus, the regularization term has the following form:

$$\Omega(U, V) = \lambda_1 \|U\|_F^2 + \lambda_2 \|V\|_{2,1}, \tag{4.12}$$

where $\|V\|_{2,1} = \sum_{i=1}^{d} \|V_{i,:}\|_2$, λ_1 and λ_2 are the tuning parameters.

Then, the proposed objective is as follows:

$$\begin{aligned} \min_{U,V,G} \quad & \eta \ell^r(U,G) + (1-\eta)\ell^s(G,V) + \Omega(U,V) \\ \text{s.t.} \quad & G\mathbf{1} = \mathbf{1} \\ & G_{i,j} \in \{0,1\}, \quad i = 1,\ldots,n,\ j = 1,\ldots,K, \end{aligned} \tag{4.13}$$

where scalar η is in $(0,1)$ and controls the tradeoff between the loss of modeling and the loss of subgrouping, $\mathbf{1} \in \mathbb{R}^K$ is a column vector of all ones and the first constraint guarantees that each data point is assigned to exactly one subgroup.

4.2.2 Optimization and Prediction

The proposed formulation (4.13) is non-convex due to the coupled unknown variable U and $\mathbf{g}_i$'s and we propose to solve it via alternating minimization. That is, we minimize the objective (4.13) over variable U with V and G fixed, minimize the objective over V with U and G fixed and minimize the objective over G with U and V fixed. The optimization with respect to variable U and V is well studied in the literature. We next briefly discuss the optimization over variable G. For the problem (4.13), the constraints are imposed on each data point and therefore we can obtain G by solving the following n subproblems for unknown variable $\mathbf{g}_i$'s separately:

$$\begin{aligned} \min_{\mathbf{g}_i} \quad & \eta \left(y_i - \mathbf{x}_i^T U \mathbf{g}_i\right)^2 + (1-\eta)\left(\mathbf{g}_i - V^T \mathbf{x}_i\right)^2 \\ \text{s.t.} \quad & \mathbf{g}_i^T \mathbf{1} = 1 \\ & g_{i,j} \in \{0,1\}, \quad j = 1,\ldots,K, \end{aligned} \tag{4.14}$$

where $g_{i,j}$ represents the j-th element in $\mathbf{g}_i$. One can solve the subproblem (4.14) by enumerating all possible cases of $\mathbf{g}_i$ and picking up the one with minimum objective value.

4.3 Experiments

In this section, we conduct experiments on the Yelp Open Dataset [1] to demonstrate the effectiveness of the proposed method for simultaneous subgrouping and learning. Yelp founded in 2005 created a platform for users to rate and review local businesses. It released the Yelp Open Dataset containing a subset of business, review and user data to the public for academic purposes. This dataset offers huge opportunities for many research topics such as community detection, recommender systems, sentiment analysis and so on. We here consider a simple regression task related to sentiment analysis where the users' text reviews are used to predict the rating score of a business. In the Yelp Open Dataset, there are total $1,326,101$ user reviews in the format of text. We first preprocess the text reviews by removing punctuations, converting all characters to lowercase and removing stop words and then vectorize each user review using the term frequency-inverse document frequency (TF-IDF) (Salton and Buckley, 1988). We build the vocabulary in TF-IDF and ignore terms that have a document frequency strictly lower than 0.1%, which results in a feature vector of dimension $4,791$. We let each user to appear only once and obtain $1,060,880$ items of user reviews. The rating scores of businesses are ranging from 0 to 5.

For the regression task, the data is expected to be heterogeneous due to variety of the businesses. For instance, the reviews or descriptions of restaurants and home services could be far different. Putting them together and learning a unified rating model may not yield satisfactory performances, and a model determining the fondness of a user by only relying on words like "love" or "hate" is not desired in practice. In this experiment, we compare our proposed method with ridge regression, Lasso regression, ridge regresion using interactions (inter-Ridge), Lasso using interactions

[1] https://www.yelp.com/dataset

Table 4.1: Comparisons of rMSEs achieved on Yelp Reviews dataset by Ridge, Lasso, KM-Ridge, KM-Lasso and the proposed methods with different K's and η's.

method	Ridge	Lasso	KM-Lasso
rMSE±std	0.868 ± 0.001	0.892 ± 0.001	0.886 ± 0.004
method	KM-Ridge	inter-Lasso	inter-Ridge
rMSE±std	0.858 ± 0.002	0.888 ± 0.001	0.829 ± 0.001
method	$K = 2\ \eta = 0.9$	$K = 4\ \eta = 0.9$	$K = 5\ \eta = 0.9$
rMSE±std	0.808 ± 0.011	0.819 ± 0.010	0.825 ± 0.012
method	$K = 8\ \eta = 0.9$	$K = 12\ \eta = 0.9$	$K = 20\ \eta = 0.9$
rMSE±std	0.862 ± 0.011	0.906 ± 0.011	0.942 ± 0.011
method	$K = 3\ \eta = 0.6$	$K = 3\ \eta = 0.7$	$K = 3\ \eta = 0.8$
rMSE±std	0.849 ± 0.013	0.846 ± 0.021	0.822 ± 0.016
method	$K = 3\ \eta = 0.9$	$K = 3\ \eta = 0.95$	$K = 3\ \eta = 0.99$
rMSE±std	0.807 ± 0.013	0.820 ± 0.018	0.853 ± 0.006

(inter-Lasso) and the one first clustering with K-means and then building seperate ridge/Lasso regression models (named KM-Ridge and KM-Lasso). We conduct 5 rounds of experiments on different splits of the dataset where 5% is for training and the rest for testing. The parameters are selected by 3-fold cross validation. The parameters for ridge regression are ranging from $[0.1, 1, 10, 50, 100, 200, 500]$. The parameters for Lasso regression are chosen from $[0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05]$. The K-means runs with 5 different centroid seeds and the maximum number of it-

erations of K-means for a single run is set as 300. For our proposed method, the λ_1 is chosen from $[1, 1.5, 10, 15, 20]$ and the λ_2 is fixed to 0.1. We test the proposed method by choosing K from $[2, 3, 4, 5, 8, 12, 20]$ with the objective tradeoff parameter in Eq.(4.13) η fixed to 0.9. We also test the proposed method with η's varying from $[0.6, 0.7, 0.8, 0.9, 0.95, 0.99]$ under K is set as3. The means and standard deviations of the root of mean squared error (rMSE) obtained in 5 rounds of experiments are reported.

From the Table 4.1, we observe that ridge regression outperforms other baseline methods. The high rMSE obtained by KM-Ridge and KM-Lasso in this experiment indicating that the clustering/subgrouping and modeling independently may easily fail to find groups that behave homogeneously in prediction tasks. For fixed $\eta = 0.9$, we observe that the models with $K = 2$ and $K = 3$ achieve similar low rMSEs compared with other methods and dividing data points into 3 subgroups achieves the lowest mean rMSE. The performances are getting worse as the subgroup number K increases which indicates that segmenting data points with finer granularity vulnerates the gernerlization. When the subgroup number K is fixed to 3, we observe that the rMSE first decreases as η increases, achieves the lowest at $\eta = 0.9$, indicating that the loss of regression task is getting dominant and a good subgrouping leads to improving fitting results. As the η continues to increase, the prediction performance tend to be worse as the weight for subgrouping loss is too small to obtain a proper subgrouping. We further investigate the subgrouping results obtained by the proposed method. Each business in a review has multiple category tags. For example, a category tag may be like "Restaurants", "home service", "Brunch" etc. We first use the model obtained with $K = 3$ and $\eta = 0.9$ that shows the lowest mean rMSE in the first round of experiment to subgroup the whole dataset into 3 groups. We then calculate the frequencies of category tags of the businesses in each subgroup and

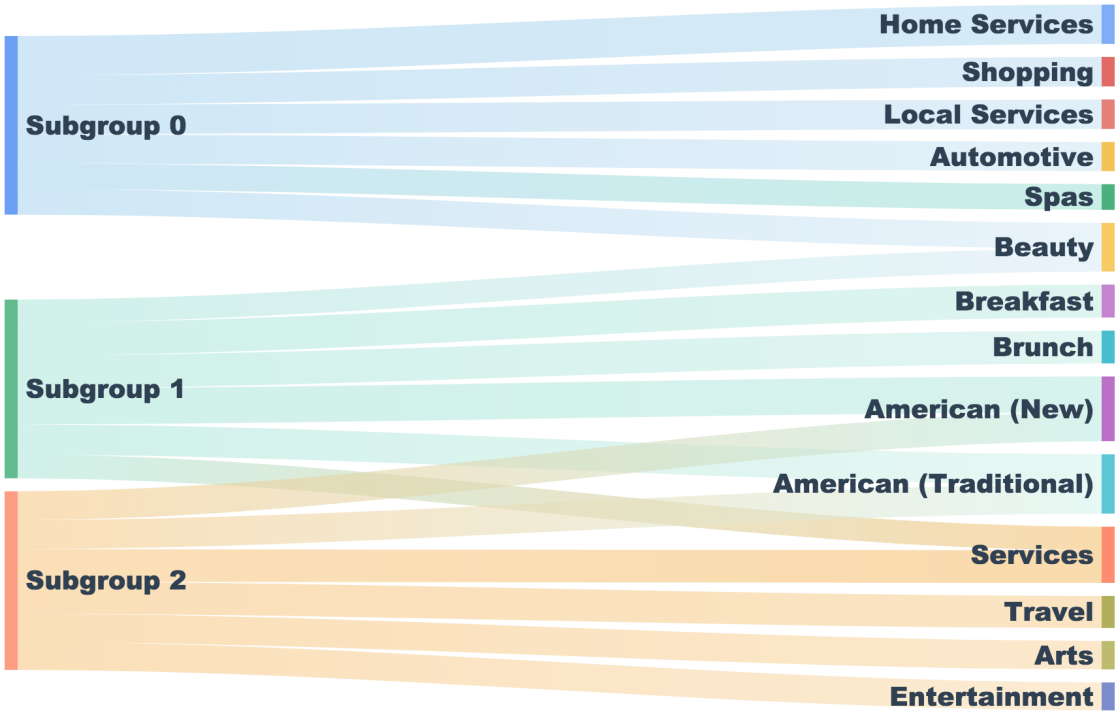

Figure 4.2: The Sankey diagram depicting the majority of categories (with common categories removed) in the three subgroups made by the proposed method.

select the top 12 frequent categories in each subgroup. In order to visualize the difference of the three subgroups, we remove the common tags appearing in all subgroups including "Restaurants", "Nightlife", "Bars", "Hotels", "Event Planning", "Food". We plot the tags distributions of the three subgroups in a Sankey diagram (Figure 4.2). The figure indicates that the three subgroups show different patterns in terms of the majority of category tags. For example, the top frequent category tags in subgroup 0 are "Home services", "Shopping", "Local Services" while tags of "Breakfast", "Brunch" appear frequently in subgroup 1 and the majority tags in subgroup 2 are "Entertainment","Arts","Travel". This observation shows that our proposed method is able to divide the data points into groups that behave differently across groups but similarly within group.

Chapter 5

CONCLUSIONS AND POSSIBLE FUTURE WORK

In this chapter, I summarize major contributions of this theis and discuss possible future works.

This thesis is built aroud the topic of mining data using models with interactions. Hierarchical Lasso methods in existing literature are proposed to achieve both sparse and hierarchical structural solutions which enables selecting important features and making models interpretable. However, the formulation of weak hierarchical Lasso is non-convex and the original work solved it by tackling a relaxed version. We first propose an efficient algorithm, eWHL, to directly solve the non-convex weak hierarchical Lasso. One critical step in eWHL is to compute the proximal operator associated with the non-convex penalty functions. As one of our major contributions, we show that the proximal operator associated with the regularization function in weak hierarchical Lasso admits a closed form solution. Furthermore, we develop an efficient algorithm which computes each subproblem of the proximal operator with a time complexity of $\mathcal{O}(d \log d)$. The technique can be easily extended to solving the strong hierarchical Lasso formulation by using non-convex ADMM methods which is worth for future explorations. Extensive experiments on both synthetic and real data sets demonstrate the superior performance of the proposed algorithm in terms of efficiency and accuracy. We then extend the non-convex formulation for the hierarchical testing and show the closed form solutions to the test statistics. The simulation studies demonstrate the superiority of the proposed non-convex hierarchical testing framework. Extending the non-convex weak hierarchical Lasso and hierarchical testing methods to other challenging applications such as depression study (Liu *et al.*, 2013)

is an important future work.

We then concentrate on modeling drug discovery problems with bi-linear models which predict the label of a data pair using feature interactions from the two data points. Specifically, we first propose a novel framework named *Dyadic Positive-Unlabeled* learning that ranks positive drug-disease/drug-drug interactions at the top. Different from most existing methodologies that treat unlabeled interactions as negatives, the proposed framework is able to detect more positive interactions by forcing the scores of positives to rank above the average score of unlabeled samples. Moreover, we derive the dual formulation of the proposed framework with the rectifier scoring function and show that the associated proximal operator admits a closed form solution. We conduct extensive experiments on real datasets and the experimental results show that our proposed framework achieves superior predictive performance compared with the state-of-the-art methods. Our method could help identify drug repositioning opportunities and predict potentially hazardous drug interactions, which will benefit patients by offering more effective and safer treatments. We further generalize the idea of "ranking above average" to "ranking above the top-ranked" by proposing a novel robust algorithm, named SortPush. We show that the proposed formulation can be efficiently solved using the ADMM framework. We show that the multivariate dual problem of the non-smooth constrained subproblem in ADMM can be converted to a one-dimensional concave maximization problem that can be efficiently solved via binary search. We demonstrate the effectiveness of SortPush against several baseline ranking at the top models on large-scale benchmark datasets. Our numerical study shows strong evidence that the sorted weighting is critical in designing a well-performed bipartite ranking model. It would be interesting to design a smarter adaptive weighting scheme in the SortPush under various noise distribution oracles and we leave these as open problems for future research.

At the end of this thesis, we propose a framework that enables simultaneous subgrouping and learning for heterogeneous data motivated by the interpretation of models using feature interactions. Compared to the classic clusterwise regression, our proposed model-based framework is free from the problem of assigning an unseen data point to a group, and is able to identify important variables critical to subgrouping and thus discover extra knowledge. We conduct empirical studies on user reviews dataset and the results show that the proposed method is able to find homogeneous subgroups and beat baseline methods in terms of generalization performance. One future research direction is to investigate multi-task learning techniques that can be embedded in the proposed framework to explore relatedness among the simultaneous learned subgroups for knowledge mining and improving generalization performance.

Adams, C. P. and V. V. Brantner, "Estimating the cost of new drug development: is it really $802 million?", Health Affairs **25**, 2, 420–428 (2006).

Agarwal, S., "The infinite push: A new support vector ranking algorithm that directly optimizes accuracy at the absolute top of the list.", in "SDM", pp. 839–850 (SIAM, 2011).

Atias, N. and R. Sharan, "An algorithmic framework for predicting side effects of drugs", Journal of Computational Biology **18**, 3, 207–218 (2011).

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

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

Bien, J., J. Taylor and R. Tibshirani, "A lasso for hierarchical interactions", The Annals of Statistics **41**, 3, 1111–1141 (2013).

Bien, J. and R. Tibshirani, *hierNet: A Lasso for Hierarchical Interactions*, URL `http://CRAN.R-project.org/package=hierNet`, r package version 1.5 (2013).

Bogdan, M., E. v. d. Berg, W. Su and E. Candes, "Statistical estimation and testing via the sorted l1 norm", arXiv preprint arXiv:1310.1969 (2013).

Boyd, S., C. Cortes, M. Mohri and A. Radovanovic, "Accuracy at the top", in "Advances in neural information processing systems", pp. 953–961 (2012).

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

Boyd, S. and L. Vandenberghe, *Convex optimization* (Cambridge university press, 2004).

Burges, C., R. Ragno and Q. Le, "Learning to rank with non-smooth cost functions", in "NIPS", (2007).

Burges, C., T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton and G. Hullender, "Learning to rank using gradient descent", in "Proceedings of the 22nd international conference on Machine learning", pp. 89–96 (2005).

Cadoret, R. J., W. R. Yates, G. Woodworth, M. A. Stewart *et al.*, "Genetic-environmental interaction in the genesis of aggressivity and conduct disorders", Archives of General Psychiatry **52**, 11, 916 (1995).

Candes, E. J. and J. Romberg, "Quantitative robust uncertainty principles and optimally sparse decompositions", Foundations of Computational Mathematics **6**, 2, 227–254 (2006).

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

Chakrabarti, S., R. Khanna, U. Sawant and C. Bhattacharyya, "Structured learning for non-smooth ranking losses", in "ACM SIGKDD", pp. 88–96 (2008).

Chapelle, O. and S. S. Keerthi, "Efficient algorithms for ranking with svms", Information Retrieval **13**, 3, 201–215 (2010).

Chapelle, O., Q. Le and A. Smola, "Large margin optimization of ranking measures", in "NIPS Workshop: Machine Learning for Web Search", (2007).

Chapelle, O., V. Vapnik, O. Bousquet and S. Mukherjee, "Choosing multiple parameters for support vector machines", Machine learning **46**, 1-3, 131–159 (2002).

Chapelle, O. and M. Wu, "Gradient descent optimization of smoothed information retrieval metrics", Information retrieval **13**, 3, 216–235 (2010).

Chipman, H., "Bayesian variable selection with related predictors", Canadian Journal of Statistics **24**, 1, 17–36, URL `https://onlinelibrary.wiley.com/doi/abs/10.2307/3315687` (1996).

Choi, N. H., W. Li and J. Zhu, "Variable selection with the strong heredity constraint and its oracle property", Journal of the American Statistical Association **105**, 489, 354–364 (2010).

Christakopoulou, K. and A. Banerjee, "Collaborative ranking with a push at the top", in "WWW", (2005).

Clémençon, S. and N. Vayatis, "Ranking the best instances", The Journal of Machine Learning Research **8**, 2671–2699 (2007).

Cossock, D. and T. Zhang, "Statistical analysis of bayes optimal subset ranking", IEEE Transactions on Information Theory **54**, 11, 5140–5154 (2008).

d'Aspremont, A., L. El Ghaoui, M. I. Jordan and G. R. Lanckriet, "A direct formulation for sparse pca using semidefinite programming", in "NIPS", vol. 16, pp. 41–48 (2004).

Davatzikos, C., P. Bhatt, L. M. Shaw, K. N. Batmanghelich and J. Q. Trojanowski, "Prediction of mci to ad conversion, via mri, csf biomarkers, and pattern classification", Neurobiology of aging **32**, 12, 2322–e19 (2011).

Dawson, J. F. and A. W. Richter, "Probing three-way interactions in moderated multiple regression: development and application of a slope difference test.", Journal of Applied Psychology **91**, 4, 917 (2006).

Deodhar, M. and J. Ghosh, "A framework for simultaneous co-clustering and learning from complex data", in "Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 250–259 (ACM, 2007).

DeSarbo, W. S. and W. L. Cron, "A maximum likelihood methodology for clusterwise linear regression", Journal of Classification **5**, 2, 249–282, URL `https://doi.org/10.1007/BF01897167` (1988).

Devanand, D., G. Pradhaban, X. Liu, A. Khandji, S. De Santi, S. Segal, H. Rusinek, G. Pelton, L. Honig, R. Mayeux *et al.*, "Hippocampal and entorhinal atrophy in mild cognitive impairment prediction of alzheimer disease", Neurology **68**, 11, 828–836 (2007).

Duchi, J., S. Shalev-Shwartz, Y. Singer and T. Chandra, "Efficient projections onto the l 1-ball for learning in high dimensions", in "Proceedings of the 25th international conference on Machine learning", pp. 272–279 (ACM, 2008).

Eley, T. C., K. Sugden, A. Corsico, A. M. Gregory, P. Sham, P. McGuffin, R. Plomin and I. W. Craig, "Gene–environment interaction analysis of serotonin system markers with adolescent depression", Molecular psychiatry **9**, 10, 908–915 (2004).

Elkan, C. and K. Noto, "Learning classifiers from only positive and unlabeled data", in "Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 213–220 (ACM, 2008).

Fazel, M., H. Hindi and S. P. Boyd, "A rank minimization heuristic with application to minimum order system approximation", in "American Control Conference, 2001. Proceedings of the 2001", vol. 6, pp. 4734–4739 (IEEE, 2001).

Fennema-Notestine, C., D. J. Hagler, L. K. McEvoy, A. S. Fleisher, E. H. Wu, D. S. Karow and A. M. Dale, "Structural mri biomarkers for preclinical and mild alzheimer's disease", Human brain mapping **30**, 10, 3238–3253 (2009).

Freund, Y., R. Iyer, R. E. Schapire and Y. Singer, "An efficient boosting algorithm for combining preferences", The Journal of machine learning research **4**, 933–969 (2003).

Gatt, J., C. Nemeroff, C. Dobson-Stone, R. Paul, R. Bryant, P. Schofield, E. Gordon, A. Kemp and L. Williams, "Interactions between bdnf val66met polymorphism and early life stress predict brain and arousal pathways to syndromal depression and anxiety", Molecular psychiatry **14**, 7, 681–695 (2009).

Gonen, M. and S. Kaski, "Kernelized bayesian matrix factorization", Pattern Analysis and Machine Intelligence, IEEE Transactions on **36**, 10, 2047–2060 (2014).

Gong, P., J. Ye and C. Zhang, "Multi-stage multi-task feature learning.", in "NIPS", pp. 1997–2005 (2012).

Gong, P., C. Zhang, Z. Lu, J. Huang and J. Ye, "A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems", in "ICML", (2013a).

Gong, P., C. Zhang, Z. Lu, J. Huang and J. Ye, *GIST: General Iterative Shrinkage and Thresholding for Non-convex Sparse Learning*, Tsinghua University, URL `http://www.public.asu.edu/~jye02/Software/GIST` (2013b).

Hastie, T., R. Tibshirani, J. Friedman, T. Hastie, J. Friedman and R. Tibshirani, *The elements of statistical learning*, vol. 2 (Springer, 2009).

Hennig, C., "Models and methods for clusterwise linear regression", in "Classification in the Information Age", pp. 179–187 (Springer, 1999).

Hennig, C., *Datenanalyse mit Modellen für Cluster linearer Regression* (diplom. de, 2000a).

Hennig, C., "Identifiablity of models for clusterwise linear regression", Journal of Classification **17**, 2, 273–296 (2000b).

Herbrich, R., T. Graepel and K. Obermayer, "Large margin rank boundaries for ordinal regression", NIPS pp. 115–132 (1999).

Hopkins, A. L., "Network pharmacology: the next paradigm in drug discovery", Nature chemical biology **4**, 11, 682–690 (2008).

Hsieh, C.-J., N. Natarajan and I. S. Dhillon, "Pu learning for matrix completion.", in "Proceedings of The 32nd International Conference on Machine Learning", pp. 2445–2453 (2015).

Iyer, S. V., R. Harpaz, P. LePendu, A. Bauer-Mehren and N. H. Shah, "Mining clinical text for signals of adverse drug-drug interactions", Journal of the American Medical Informatics Association **21**, 2, 353–362 (2014).

Jain, P. and I. S. Dhillon, "Provable inductive matrix completion", arXiv preprint arXiv:1306.0626 (2013).

Jakulin, A., *Machine learning based on attribute interactions*, Ph.D. thesis, Univerza v Ljubljani (2005).

Ji, S. and J. Ye, "An accelerated gradient method for trace norm minimization", in "Proceedings of the 26th Annual International Conference on Machine Learning", ICML '09, pp. 457–464 (2009).

Joachims, T., "Optimizing search engines using clickthrough data", in "ACM SIGKDD", pp. 133–142 (2002).

Joachims, T., "A support vector method for multivariate performance measures", in "Proceedings of the 22nd international conference on Machine learning", pp. 377–384 (ACM, 2005).

Koh, K., S.-J. Kim and S. Boyd, "An interior-point method for large-scale 1-regularized logistic regression.", Journal of Machine learning research **8**, 7 (2007).

Lee, I., U. M. Blom, P. I. Wang, J. E. Shim and E. M. Marcotte, "Prioritizing candidate disease genes by network-based boosting of genome-wide association data", Genome research **21**, 7, 1109–1121 (2011).

Li, H., Y. Liu, P. Gong, C. Zhang, J. Ye, A. D. N. Initiative *et al.*, "Hierarchical interactions model for predicting mild cognitive impairment (mci) to alzheimer's disease (ad) conversion", PloS one **9**, 1, e82450 (2014a).

Li, N., R. Jin and Z.-H. Zhou, "Top rank optimization in linear time", in "Advances in Neural Information Processing Systems", pp. 1502–1510 (2014b).

Liu, B., *Web data mining: exploring hyperlinks, contents, and usage data* (Springer Science & Business Media, 2007).

Liu, B., Y. Dai, X. Li, W. S. Lee and P. S. Yu, "Building text classifiers using positive and unlabeled examples", in "Data Mining, 2003. ICDM 2003. Third IEEE International Conference on", pp. 179–186 (IEEE, 2003).

Liu, Y., Z. Nie, J. Zhou, M. Farnum, V. A. Narayan, G. Wittenberg and J. Ye, "Sparse generalized functional linear model for predicting remission status of depression patients", in "Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing", vol. 19, pp. 364–375 (World Scientific, 2013).

Llano, D. A., G. Laforet and V. Devanarayan, "Derivation of a new adas-cog composite using tree-based multivariate analysis: prediction of conversion from mild cognitive impairment to alzheimer disease", Alzheimer Disease & Associated Disorders **25**, 1, 73–84 (2011).

Luo, H., P. Zhang, H. Huang, J. Huang, E. Kao, L. Shi, L. He and L. Yang, "Ddi-cpi, a server that predicts drug–drug interactions through implementing the chemical–protein interactome", Nucleic acids research p. gku433 (2014).

Montgomery, D. C., E. A. Peck and G. G. Vining, *Introduction to linear regression analysis*, vol. 821 (Wiley, 2012).

Muruzábal, J., D. Vidaurre and J. Sánchez, "Somwise regression: a new clusterwise regression method", Neural Computing and Applications **21**, 6, 1229–1241, URL `https://doi.org/10.1007/s00521-011-0536-3` (2012).

Narasimhan, H. and S. Agarwal, "A structural SVM based approach for optimizing partial AUC", in "ICML", pp. 516–524 (2013a).

Narasimhan, H. and S. Agarwal, "Svmpauctight: A new support vector method for optimizing partial auc based on a tight convex upper bound", in "ACM SIGKDD", pp. 167–175 (2013b).

Natarajan, N. and I. S. Dhillon, "Inductive matrix completion for predicting gene–disease associations", Bioinformatics **30**, 12, i60–i68 (2014).

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

Nesterov, Y., *Introductory lectures on convex optimization*, vol. 87 (Springer Science & Business Media, 2004).

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

Petersen, R. C., "Mild cognitive impairment clinical trials", Nature Reviews Drug Discovery **2**, 8, 646–653 (2003).

Radchenko, P. and G. M. James, "Variable selection using adaptive nonlinear interaction structures in high dimensions", Journal of the American Statistical Association **105**, 492, 1541–1553 (2010).

Rockafellar, R. T., *Convex analysis*, no. 28 (Princeton University Press, 1970).

Rudin, C., "The p-norm push: A simple convex ranking algorithm that concentrates at the top of the list", The Journal of Machine Learning Research **10**, 2233–2271 (2009).

Rudin, C. and R. E. Schapire, "Margin-based ranking and an equivalence between adaboost and rankboost", The Journal of Machine Learning Research **10**, 2193–2232 (2009).

Salton, G. and C. Buckley, "Term-weighting approaches in automatic text retrieval", Information processing & management **24**, 5, 513–523 (1988).

Sellamanickam, S., P. Garg and S. K. Selvaraj, "A pairwise ranking based approach to learning with positive and unlabeled examples", in "Proceedings of the 20th ACM international conference on Information and knowledge management", pp. 663–672 (ACM, 2011).

Simon, N. and R. Tibshirani, "A permutation approach to testing interactions in many dimensions", arXiv preprint arXiv:1206.6519 (2012).

Singh-Blom, U. M., N. Natarajan, A. Tewari, J. O. Woods, I. S. Dhillon and E. M. Marcotte, "Prediction and validation of gene-disease associations using methods inspired by social network analyses", PLoS ONE **8** (2013).

Somers, M. J., "Organizational commitment, turnover and absenteeism: An examination of direct and interaction effects", Journal of Organizational Behavior **16**, 1, 49–58 (1995).

Späth, H., "Algorithm 39 clusterwise linear regression", Computing **22**, 4, 367–373 (1979).

Späth, H., "A fast algorithm for clusterwise linear regression", Computing **29**, 2, 175–181 (1982).

Späth, H., *Mathematical algorithms for linear regression* (Academic Press, 2014).

Sra, S., "Nonconvex proximal splitting: batch and incremental algorithms", arXiv preprint arXiv:1109.0258 (2011).

Srebro, N., J. Rennie and T. S. Jaakkola, "Maximum-margin matrix factorization", in "Advances in neural information processing systems", pp. 1329–1336 (2004).

Tatonetti, N. P., G. H. Fernald and R. B. Altman, "A novel signal detection algorithm for identifying hidden drug-drug interactions in adverse event reports", Journal of the American Medical Informatics Association pp. amiajnl–2011 (2011).

Taylor, M., J. Guiver, S. Robertson and T. Minka, "Softrank: optimizing non-smooth rank metrics", in "International Conference on Web Search and Data Mining", pp. 77–86 (2008).

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

Tsochantaridis, I., T. Joachims, T. Hofmann and Y. Altun, "Large margin methods for structured and interdependent output variables", in "Journal of Machine Learning Research", pp. 1453–1484 (2005).

Tusher, V. G., R. Tibshirani and G. Chu, "Significance analysis of microarrays applied to the ionizing radiation response", Proceedings of the National Academy of Sciences **98**, 9, 5116–5121 (2001).

Usunier, N., D. Buffoni and P. Gallinari, "Ranking with ordered weighted pairwise classification", in "ICML", pp. 1057–1064 (2009).

Wei, W.-Q., R. M. Cronin, H. Xu, T. A. Lasko, L. Bastarache and J. C. Denny, "Development and evaluation of an ensemble resource linking medications to their indications", Journal of the American Medical Informatics Association **20**, 5, 954–961 (2013).

Wishart, D. S., C. Knox, A. C. Guo, D. Cheng, S. Shrivastava, D. Tzur, B. Gautam and M. Hassanali, "Drugbank: a knowledgebase for drugs, drug actions and drug targets", Nucleic acids research **36**, suppl 1, D901–D906 (2008).

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

Wu, X., R. Jiang, M. Q. Zhang and S. Li, "Network-based global inference of human disease genes", Molecular systems biology **4**, 1 (2008).

Xu, J. and H. Li, "Adarank: a boosting algorithm for information retrieval", in "ACM SIGIR", pp. 391–398 (2007).

Yan, L., W.-j. Li, G.-R. Xue and D. Han, "Coupled group lasso for web-scale ctr prediction in display advertising", in "International Conference on Machine Learning", pp. 802–810 (2014).

Ye, J., M. Farnum, E. Yang, R. Verbeeck, V. Lobanov, N. Raghavan, G. Novak, A. DiBernardo, V. A. Narayan *et al.*, "Sparse learning and stability selection for predicting mci to ad conversion using baseline adni data", BMC neurology **12**, 1, 46 (2012).

Yuan, M., V. R. Joseph and H. Zou, "Structured variable selection and estimation", The Annals of Applied Statistics pp. 1738–1757 (2009).

Yue, Y., T. Finley, F. Radlinski and T. Joachims, "A support vector method for optimizing average precision", in "ACM SIGIR", pp. 271–278 (2007).

Zhang, B., "Regression clustering", in "Third IEEE International Conference on Data Mining", pp. 451–458 (2003).

Zhao, P., G. Rocha and B. Yu, "The composite absolute penalties family for grouped and hierarchical variable selection", The Annals of Statistics **37**, 6A, 3468–3497, URL `http://dx.doi.org/10.1214/07-AOS584` (2009).

Zheng, X., H. Ding, H. Mamitsuka and S. Zhu, "Collaborative matrix factorization with multiple similarities for predicting drug-target interactions", in "Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 1025–1033 (ACM, 2013).

Zhou, J., J. Liu, V. A. Narayan and J. Ye, "Modeling disease progression via multi-task learning", NeuroImage **78**, 233–248 (2013).

Zou, H., T. Hastie and R. Tibshirani, "Sparse principal component analysis", Journal of computational and graphical statistics **15**, 2, 265–286 (2006).