High Dimensional Learning with
Structure Inducing Constraints and Regularizers

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Amir Asiaee Taheri

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

Arindam Banerjee

Augest, 2017

Acknowledgements

I cannot express enough thanks to my advisor, Prof. Arindam Banerjee, for his guidance, and encouragement throughout my graduate study. I am ineffably indebted to him for his continued support during my health crisis. I will forever be grateful to him for introducing me to machine learning, inspiring me in my intellectual quest, and patiently educating me along the way.

I would like to forward my gratitude towards Prof. Rui Kuang, Prof. Zhi-Li Zhang, and Prof. Susan Wei for being my dissertation committee members. Moreover, I want to thank all the professors with whom I have interacted, within and outside classes, over the years. I would also like to especially thank my collaborators Golshan Golnari, Soumyadeep Chatterjee, Mariano Tepper, Fernando Silveira, and Prof. Guillermo Sapiro.

During my years in grad school, I enjoyed the companionship of many friends and labmates at the university. My heartfelt thanks to Hamid, Mojtaba, Golshan, Farideh, Igor, Karthik, Soumyadeep, Huahua, Puja, Rudy, Vidyashankar, Konstantina, Sheng, Nick, and Andre. I enjoyed all of our discussions and learned from your insights. Also, warm thanks to my incredible friends who made Minneapolis weather tolerable! Thanks to Hamid, Masoud, Saber, Mohammad, and Abbas.

Thanks will not do justice to the unimaginable sacrifices my parents, Zarrindokht and Gholamreza, made for my success and prosperity. I am and will always be grateful for their lifelong support and encouragement. I am thankful for having a kind and compassionate sister, Elaheh with whom I share most pleasant memories. I would like to thanks her and my brother-in-law Amir Zarinbal for their immense support for my wife and me during the past two years while I was fighting cancer. Last but not least, I thank my wife Fatemeh wholeheartedly for all her love, patience, and care during our past ten years together. I could not survive the grad school without you, and I am a better person because of you.

Dedication

To my parents, Zarrindokht Davoodian and Gholamreza Asiaee, my pillars of strength
To my sister, Elaheh Asiaee, whose blood flows through my veins
To my wife, Fatemeh Khodaei, who is my angel

Abstract

Explosive growth in data generation through science and technology calls for new computational and analytical tools. To the statistical machine learning community, one major challenge is the data sets with dimensions larger than the number of samples. Low sample-high dimension regime violates the core assumption of most traditional learning methods. To address this new challenge, over the past decade many high-dimensional learning algorithms have been developed.

One of the significant high-dimensional problems in machine learning is the linear regression where the number of features is greater than the number of samples. In the beginning, the primary focus of high-dimensional linear regression literature was on estimating sparse coefficient through l_1-norm regularization. In a more general framework, one can assume that the underlying parameter has an intrinsic "low dimensional complexity" or *structure*. Recently, researchers have looked at structures beyond sparsity that are induced by *any norm* as the regularizer or constraint.

In this thesis, we focus on two variants of the high-dimensional linear model, i.e., data sharing and errors-in-variables where the structure of the parameter is captured with a suitable norm. We introduce estimators for these models and study their theoretical properties. We characterize the sample complexity of our estimators and establish non-asymptotic high probability error bounds for them. Finally, we utilize dictionary learning and sparse coding to perform Twitter sentiment analysis as an application of high dimensional learning.

Some discrete machine learning problems can also be posed as constrained set function optimization, where the constraints induce a structure over the solution set. In the second part of the thesis, we investigate a prominent set function optimization problem, the social influence maximization, under the novel "heat conduction" influence propagation model. We formulate the problem as a submodular maximization with cardinality constraints and provide an efficient algorithm for it. Through extensive experiments on several large real and synthetic networks, we show that our algorithm outperforms the well-studied methods from influence maximization literature.

Contents

List of Tables

List of Figures

Part I

Preliminaries

Chapter 1

Introduction

In recent years fields in science and technology have witnessed a rapid growth in data acquisition rate. On the scientific front, we are collecting an unprecedented amount of data every day to either conduct exploratory data analysis or refine existing theories. For example, the rate of data produced at the Large Hadron Collider for a single "collision event" is 25 gigabytes per second [2]. In the production industries, companies record minuscule interactions of the users with their products through high-resolution sensors and analyze it to improve user experience. Smart appliances, cars, buildings, and cities are few examples where the product-oriented industries attempt to exploit the stored data to increase efficiency and economic benefit [3]. Online service-oriented industries like search engines and social networks track the finest details of users' activities with the goal of personalized content delivery [4, 5].

The central limit theorem teaches us that a larger sample size leads to a more accurate estimation [6]. It is therefore not unreasonable to assume that larger data sets will pose no challenges to the classical statistical learning procedures. On the contrary, most of the modern data sets introduce new challenges for the field of statistics. Although the *number of acquired samples* n increased substantially, so too has the number of measurement per sample, known as the *problem dimension* p. And if p grows faster than n this violates a core assumption of statistical learning, namely, $n \geq p$ [7, 8]. For example, for a rare disease, one can only recruit a handful of patients to participate in a trial, while the number of measurements, e.g., genome sequence, can easily exceed few thousands. Scientific fields which use high resolution images have the same difficulty. The number of samples in any study will hardly surpass the millions of pixels measured in each picture. The regime of $p \gg n$ is the point where traditional statistical learning fails to provides us with reliable inference and prediction tools [7, 8].

Over the past decade, many novel tools have been developed to address challenges corresponding to high-dimensional data [7, 8, 9, 10, 11, 12]. All of these contemporary methods make extra assumptions to circumvent the $p \gg n$ condition. The most basic form of the assumptions can be summarized as follows: For a given problem with dimension p, only $s \ll p$ of the measured features are relevant, and the rest are simply noise. This simple assumption has evolved into a more general notion of low complexity *"structure"* for problems in high dimensions [7, 8, 13, 14, 12].

In this thesis, we focus on learning in finite dimensional parametric representations which reduces to a statistical estimation problem. We investigate a set of important problems under the assumption of $p \gg n$ and provide estimation procedures for each of them. The set of problems comprises both continuous and discrete tasks. We analyze the statistical properties of the proposed learning algorithms, such as sample complexity and estimation error bound and confirm our analysis with experimental results. In remainder of Chapter 1, we review the discrete and continuous problems studied in this manuscript (Section 1.1), survey the current state-of-the-art structures that make learning in high-dimensional regime possible (Section 1.2) and finally summarize our contributions (Section 1.3).

1.1 Studied High Dimensional Problems

The focus of this manuscript is high dimensional problems that can be posed as statistical estimation where the parameter of interest has low complexity structure. Table 1.1 summarizes the studied problems of this thesis. We formulate these estimation problems as constrained and regularized optimizations. We study three important examples of continuous objective-continuous constraints [1, 15, 16] and one prominent instance from discrete objective-discrete constraints class of optimizations [17]. For the recent advances in continuous objective-discrete constraints and other interesting connections, we refer the readers to [18, 19, 20].

More specifically, we investigate two variants of structured high dimensional linear regression, namely data sharing [15, 21] and errors-in-variables [22, 16] and provide efficient estimator for them. We explore theoretical properties of our estimators and show their statistical consistency. In addition to our theoretical contribution, we also study an application of dictionary learning [1, 23] and a discrete problem from submodular set function maximization with

structure space	continuous	discrete
continuous	Data sharing [15] Noisy linear model [16] Dictionary learning [1]	–
discrete	–	Subm max. with cardinality const. [17]

Table 1.1: Thesis contribution based on parameter space and structure.

cardinality constraint [17, 24]. In the following, we review the necessary basics of the linear model and submodular functions and introduce the studied problems in more details.

1.1.1 High Dimensional Linear Regression

Classic linear regression [6, 25] assumes the following simple linear model for observations:

$$y_i = \langle \mathbf{x}_i, \boldsymbol{\beta} \rangle + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim N(0, \sigma^2). \tag{1.1}$$

The Maximum Likelihood Estimator (MLE) of this model is the following optimization known as the Ordinary Least Squares (OLS):

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{2n} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2. \tag{1.2}$$

Constrained and regularized version of (1.2) originally have been used as a tool for model selection and to prevent overfitting. The well-known ridge regression, is a classic example of the regularized linear regression [25]:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{2n} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_2^2. \tag{1.3}$$

where $\mathbf{X} \in \mathbb{R}^{n \times p}$.

Under the assumption of high dimension, i.e., $p \gg n$, we study three variants of linear model (1.1) in Chapters 3 - 5 and provide OLS-like estimator for them. We focus on the variants of OLS objective (1.2) where instead of the l_2-norm regularization of the ridge regression (1.3), we have other norms which impose the desired structure over the parameter of interest $\boldsymbol{\beta}$. More details about these structure inducing norms will be discussed in Section 1.2. In the following we briefly introduce the variants of the linear model studied in this manuscript.

Data Sharing

Consider the problem of linear regression in high dimension when there are more than one cohort/group in the population. One can ignore this side information and model the outcome of interest using the following simple linear model:

$$y_i = \mathbf{x}_i \boldsymbol{\beta}^* + w_i.$$

On the other extreme, we can assume that for each group, the data comes from a different linear model with distict parameter $\boldsymbol{\beta}_g^*$:

$$y_{gi} = \mathbf{x}_{gi} \boldsymbol{\beta}_g^* + w_{gi},$$

where g and i index the group and samples of each group respectively.

Here we take a middle ground that has been recently suggested in the literature [21, 26, 27, 28], and assume that output data y_{gi} are coming from distinct linear models, but the parameters are not radically different. To capture this notion, one can assume that there is a shared parameter $\boldsymbol{\beta}_0^*$ between all groups which expresses the similarity between groups and a private per-group parameter $\boldsymbol{\beta}_g$ that captures the differences between the groups:

$$y_{gi} \quad = \quad \mathbf{x}_{gi}(\boldsymbol{\beta}_0^* + \boldsymbol{\beta}_g^*) + w_{gi}, \quad g \in \{1, \dots, G\}. \tag{1.4}$$

In (1.4), we have G linear regression that share data through $\boldsymbol{\beta}_0^*$. We call these set of models, *"data sharing"* model. Here, we are interested in estimation of both shared and private parameters in the high dimensional regime.

Error-in-variabls

The study of regression models with errors in features predates the twentieth century [29]. In the simplest setting for such models, we assume that instead of observing $(\mathbf{x}_i, y_i)$ from the linear model $y_i = \langle \boldsymbol{\beta}^*, \mathbf{x}_i \rangle + \epsilon_i$, $(\mathbf{z}_i, y_i)$ is observed, where $\mathbf{z}_i = f(\mathbf{x}_i, \mathbf{w}_i)$ is a noisy version of $\mathbf{x}_i$ corrupted by $\mathbf{w}_i$. The form of function f which we consider in this manuscript is additive noise. Hence, our noisy measurement model of interest is:

$$\begin{aligned} y_i &= \langle \boldsymbol{\beta}^*, \mathbf{x}_i \rangle + \epsilon_i, \quad \boldsymbol{\beta}^* \in \mathbb{R}^p && (1.5) \\ \mathbf{z}_i &= \mathbf{x}_i + \mathbf{w}_i. && (1.6) \end{aligned}$$

In this thesis, our goal is to estimate the parameter $\boldsymbol{\beta}^*$ in high dimensional regime, given the noisy observations.

Dictionary Learning for Sentiment Classification

The goal of sentiment analysis is to classify a text's "emotion" as positive or negative. One can conduct opinion polls using sentiment analysis, in a natural and *non-intrusive* way by monitoring social media about a given topic and analyzing the sentiments of the content [30, 31]. Much interesting work has been done on Twitter's sentiment analysis. Detecting major events based on tweets' sentiments [32, 31, 33], finding pattern of temporal happiness and mood in human behavior [34, 35] are only some applications of the sentiment analysis for Twitter data.

In this manuscript, we represent each tweet as a vector $\mathbf{t}_i$ which is made out of either "happy" or "sad" mood matrices (dictionaries), $\mathbf{D}_{\text{happy}}$ and $\mathbf{D}_{\text{sad}}$, as follows:

$$\mathbf{t}_i = \mathbf{D}_{\text{mood}} \boldsymbol{\beta}^*_{\mathbf{t}_i} + \boldsymbol{\epsilon}_i \tag{1.7}$$

where mood is happy or sad and $\boldsymbol{\beta}^*_{\mathbf{t}_i}$ is called the corresponding code of the tweet $\mathbf{t}_i$. We want to solve (1.7) in the high dimensional regime, i.e., when $\mathbf{D}_{\text{mood}}$ is a fat matrix. In contrast to previous problems, here both dictionaries and code should be learned from data $\{(\mathbf{t}_i, y_i)\}_{i=1}^n$ where y_i is the given happy or sad label of tweet i.

1.1.2 Submodular Maximization

Over the past decade set function optimization has become an important part of the machine learning literature [19, 36]. Sensor placement [37], the value of information [38] and influence maximization in social networks [17, 24] are just a few examples of problems that can be formulated as constrained set function optimization where the constraints induce structure on the solution. In its most general form, the set function optimization can be posed as follows:

$$\hat{\mathcal{S}} = \underset{\mathcal{S} \in 2^{\mathcal{U}}}{\operatorname{argmin}}\, \sigma(\mathcal{S}) \quad \text{s.t.} \quad \mathcal{S} \in \mathcal{V}, \tag{1.8}$$

where $\mathcal{U}$ is the universal set, $\mathcal{V} \subset 2^{\mathcal{U}}$ is the constraint set and σ is the set function of interest. The simplest constraint can be imposed on the cardinality of the set $\mathcal{V}$ as $|\mathcal{V}| \leq k$. Cardinality constraint follows the core assumption of high dimensional modeling, i.e., a small number of set members has a large effect on the function of interest σ.

The sensor placement problem is a classic example where we are interested in deploying s fire alarms to cover an as large area as possible. There are p possible deployment sites (dimension of the problem) and we should choose s of them as the set $\mathcal{S}$ (sparsly selected deployment area), which has the maximum coverage measured by the set function $\sigma(\mathcal{S})$ [36]:

$$\hat{\mathcal{S}} = \underset{\mathcal{S}:|\mathcal{S}|\leq s}{\operatorname{argmax}}\, \sigma(\mathcal{S}). \tag{1.9}$$

A property of function σ that makes optimizations like (1.9) tractable is *submodularity* [19, 36]. Submodular functions in discrete domains, similar to the convex function in continuous domains, are easier to optimize.

Heuristically, submodularity is a diminishing return property. In our previous example, as you add more fire detectors, the marginal coverage of the new detectors diminishes, simply because the chance of covering already covered regions increases. We say that the *"marginal gain"* of adding more detectors decreases. We call this property *"diminishing return"*, meaning that enlarging the selected set has diminishing benefit for the optimization objective. Similar to the sensor placement example, diminishing return property holds true for many real world problems and translates to a beautiful mathematical property of *submodularity*.

We study the problem of social influence maximization in Chapter 6 and formulate it as a submodular function maximization under the cardinality constraints. In the following, we briefly introduce this problem, and leave more details for Chapter 6.

Influence Maximization

Motivated by viral marketing [39] and other applications, the problem of influence maximization in social networks has attracted much attention in recent years. In its basic form, we are interested in finding a small set of individuals to target for maximizing the spread of a new product adoption. In this setting, we measure the influence of the set $\mathcal{S}$ of customers with the influence function $\sigma(\mathcal{S})$ which is the expectation of the number of customers that will purchase a product due to the word-of-mouth phenomena. We want to maximize the influence function under the assumption that a small set of customers with size s can have a major effect on product sales:

$$\hat{\mathcal{S}} = \underset{\mathcal{S}:|\mathcal{S}|\leq s}{\operatorname{argmax}}\, \sigma(\mathcal{S}) \tag{1.10}$$

Some basic properties of the influence function σ are determined from sociology, psychology, economics and game theory. Any function that matches those properties is a potential candidate for being an influence function. The mathematical nature of the assumed influence function σ will affect computational complexity of the algorithm needed for solving (1.10). Here, we introduce a new influence function that captures "changes in loyalty of customers" and under this formulation we solve (1.10).

1.2 Structure Inducing Constraints

Following our introduction of the problems of interest, in this section, we elaborate on possible structures of the parameters involved in those problems. We first introduce variants of the classic sparsity in Section 1.2.1. Then in Section 1.2.2 we discuss how these variants are formulated as a constraint in an optimization, which leads us to newer forms of structures discussed in Section 1.2.3.

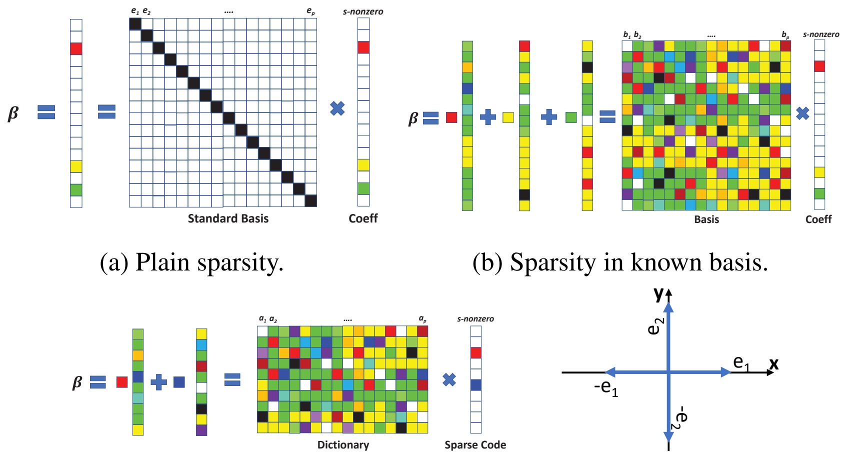
1.2.1 Classic Sparsity

Sparsity can be modeled as l_0-pseudo-norm constraint in our optimization. Formally, l_0-pseudo-norm counts the number of non-zeros in vector $\boldsymbol{\beta}$, i.e., $\|\boldsymbol{\beta}\|_0 = |\text{Supp}(\boldsymbol{\beta})|$. In general the sparsity structure of $\boldsymbol{\beta}$ can be richer than just "small support". In *classic sparsity*, the parameter $\boldsymbol{\beta}$ can have a sparse representation in another basis. Here we elaborate on variants of the classic notion of sparsity.

Plain Sparsity. As mentioned, the simplest assumption of high dimensional statistics is that out of p possible features or dimensions only s of them are relevant to a problem. In a parametric setting, this means that the parameter of interest $\boldsymbol{\beta}$ is s-sparse, or in other words the cardinality of its support is s, $|\text{Supp}(\boldsymbol{\beta})| = s$ [7]. Therefore, out of all p possible *standard bases* $\{\mathbf{e}_i\}_{i=1}^p$, only s of them contribute in making $\boldsymbol{\beta}$, Figure 1.1a.

Sparsity in Known Basis. A parameter or signal of interest may be sparse in a basis other than the standard basis. For example, media data like still imagery, video, and acoustic data can be sparsely represented using other bases like Fourier or wavelet [40], Figure 1.1b.

Sparsity in Learned Dictionary. To get a sparse representation, instead of using the known basis, sometimes it is more helpful to learn a set of basis from data. Since the goal is a sparse representation, usually the set of basis becomes over-complete, which means that the elements of the basis set are not linearly independent. Therefore, mathematically speaking the set is not a basis. For this reason, each learned vector is called an *atom* and the collection of atoms form



(a) Plain sparsity.

(b) Sparsity in known basis.

(c) Learned over-complete dictionary.

(d) Set of atoms.

Figure 1.1: Illustration of classic sparsity.

a matrix known as *dictionary*, Figure 1.1c. Note that dictionary learning and sparse coding are two intertwined problems and are usually formulated as an alternate optimization [41].

Sparsity in a Set of Atoms. This setting has similarity with both previous cases: the parameter can be sparsely represented by a (possibly) overcomplete set of discrete predefined atoms $\mathcal{A}$ [42]. The set of atoms $\mathcal{A}$ is determined by the application of interest. Consider $\mathcal{A} = \{\pm \mathbf{e}_i\}_{i=1}^n$ as the set of atoms. Set $\mathcal{A}$ is overcomplete (similar to dictionaries) but predefined (similar to known basis), Figure 1.1d.

1.2.2 From Structures to Norms

As mentioned earlier, searching for the s most relevant features is a combinatorial and intractable problem when solved exactly by exhaustive search. We can formulate this combinatorial problem as an l_0-pseudo-norm constraint optimization. Here is the example for OLS:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}:\|\boldsymbol{\beta}\|_0 \leq s}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2, \tag{1.11}$$

The optimal solution of (1.11) is an s-sparse vector that minimizes the OLS loss function. The closest convex relaxation of this combinatorial constraint is known to be the l_1-norm which leads to the following program:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}:\|\boldsymbol{\beta}\|_1 \leq t}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2. \tag{1.12}$$

(1.12) is the constraint form of the well-known LASSO [9] estimator. Note that the constant t in the problem (1.12) is different form s in the original problem (1.11) and in practice is determined by cross validation. We can write the regularized version of (1.12) for OLS loss as:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{2n} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1, \tag{1.13}$$

which is the LASSO estimator. LASSO and its variants were very successful in practice, but beyond some intuitive explanations, little were known about the conditions under which they were guaranteed to work. Over the past decade, a body of literature has developed that precisely characterizes where and why LASSO or similar estimators will recover the actual sparse solution [11, 12, 43, 42].

The idea of convexifying a constraint that constructed (1.12) from (1.11) has been used to extend possible parameter structure beyond sparsity [42]. For example, when we have sparsity in a set of atoms 1.2.1, the convex hull of $\mathcal{A}$, induces a norm known as *atomic norm* which can be used for efficient recovery of $\boldsymbol{\beta}$ via a convex program. In the case of $\mathcal{A} = \{\pm\mathbf{e}_i\}_{i=1}^n$, the atomic norm is the well-known l_1 norm. The loss convexification idea leads us to a more recent method of capturing structure which is based on general norms.

1.2.3 Structures Beyond Sparsity

In this work, $\boldsymbol{\beta}$ is structured if it has small value due to a suitable function that captures the structure. Often, these functions are either norms of $\boldsymbol{\beta}$ or submodular function over the Supp($\boldsymbol{\beta}$). Norms are convex functions and can be used directly in a convex program, but for structures based on submodular functions, one can derive a corresponding surrogate norm and add it as a constraint to the optimization [18].

Structure Induced by Norms. Recently, it is proposed to use *any norm* to capture more complex structures [14, 12]. So any real function $R(\cdot)$ that satisfies the three basic properties of norms can be used to model the structure of the parameter. A parameter $\boldsymbol{\beta}$ is structured due to the norm $R(\cdot)$ if $R(\boldsymbol{\beta})$ is small. Note that, norm induced structures generalize atomic norms (convex hull of a set of atoms), which itself is a generalization of the plain sparsity.

Finally, one can go one step further and consider $\boldsymbol{\beta}$ structured if it belongs to a *feasible set*, $\boldsymbol{\beta} \in \mathcal{F}$. For example, if $\boldsymbol{\beta}$ is structured due to a norm $R(\cdot)$ with the value $R(\boldsymbol{\beta})$, then $\boldsymbol{\beta} \in R(\boldsymbol{\beta})\Omega_R$ where Ω_R is the unit norm ball of $R(\cdot)$. Although our analysis readily extends from norm balls to general feasible sets, our focus here is the structures induced by norms.

Structure Induced by Submodular Function. For the class of discrete optimization problems, constraints can be any feasible sets but submodular constraints are easier to deal with [44]. For example, in the maximum coverage problem (1.9) both the objective (the coverage function $\sigma(\mathcal{S})$) and the cardinality constraint are submodular.

In problems with a continuous objective, many types of aforementioned structures in Section 1.2.1 can also be captured by submodular set functions over the support of the parameter $\text{Supp}(\boldsymbol{\beta})$. Therefore, the objective is continuous but constraints are discrete. To efficiently solve the optimization problem, one should convert it to a convex program by using the continuous convex envelope of the constraint submodular functions as the relaxed convex constraint [18].

1.3 Contributions

In this section, we briefly introduce problems studied as part of this thesis and our contributions.

1.3.1 High Dimensional Data Sharing

We study the data sharing model of (1.4) when number of dimension is larger than number of samples. In high dimensional regime, we assume that both shared and private parameters are structured, i.e., for suitable norms, $R_g(\boldsymbol{\beta}_g^*)$s are small. For example, when the structure is sparsity the corresponding norm is l_1-norm and one desirable scenario is when the shared parameter is much denser than the private parameters. In other words, for s_g-sparse $\boldsymbol{\beta}_g^*$s we have $s_0 \gg s_g$. The shared parameter expresses the "dense similarity" and private parameters capture "slight difference" between groups.

We propose an estimator for recovering the structured shared and private parameters where the structure is induced by norms $R_g(\cdot)$. We derive the following results:

- We show high probability non-asymptotic bound on the weighted sum of component-wise estimation error, $\boldsymbol{\delta}_g = \hat{\boldsymbol{\beta}}_g - \boldsymbol{\beta}_g^*$ as:

  $$\sum_{g=0}^{G} \alpha_g \|\boldsymbol{\delta}_g\|_2 \leq c \frac{\max_{g \in [G]} \omega(\mathcal{C}_g \cap \mathbb{S}^{p-1}) + \sqrt{\log G}}{\sqrt{n}}, \quad \alpha_g = \sqrt{\frac{n_g}{n}}, [G] = \{0, \dots, G\} \tag{1.14}$$

  where n_g is number of samples per group, $n = n_0$ is the total number of samples, and Gaussian width of a set $\mathcal{S}$ is $\omega(\mathcal{S}) = \mathbb{E}_{\mathbf{g}}\left[\sup_{\mathbf{u} \in \mathcal{S}} \langle \mathbf{g}, \mathbf{u} \rangle\right]$ [42]. Also $\mathcal{C}_g$ is the error cone corresponding to $\boldsymbol{\beta}_g^*$ exactly defined in Section 3.2.

- The general bound of (1.14) entails following bounds for specific parameters:

$$\forall g \in [G] : \|\boldsymbol{\delta}_g\|_2 \leq c \frac{\max_{g \in [G]} \omega(\mathcal{C}_g \cap \mathbb{S}^{p-1}) + c\sqrt{\log G}}{\sqrt{n_g}} \tag{1.15}$$

  It can be observed that l_2-norm of the estimation error for the shared component decays as $1/\sqrt{n}$ which is similar to the well-studied high dimensional regression case [14]. So the estimation of the shared component exploit all of the pooled data to reduce its error.

- We also show in Section 3.3 that the required sample complexity for the recovery of parameters should be simultaneously satisfied for all groups as $n_g \geq c_g(\omega(\mathcal{C}_g \cap \mathbb{S}^{p-1}) + \sqrt{\log G})^2$ and for the shared parameter as $n \geq c_0(\omega(\mathcal{C}_0 \cap \mathbb{S}^{p-1}) + \sqrt{\log G})^2$ where $\omega(\cdot)$ is the Gaussian width. In other words, enough *total* number of samples is necessary to recover the shared parameter. So we can show that the shared parameter benefits from the pooled data.

- Finally we present a fast optimization algorithm that converge linearly to the solution of our proposed estimator.

1.3.2 High Dimensional Noisy Regression

Given $\{(\mathbf{z}_i, y_i)\}_{i=1}^n$ of (1.5) and (1.6), we want to compute $\hat{\boldsymbol{\beta}}$, which is l_2 consistent, i.e., for the error vector $\boldsymbol{\delta} = \hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*$, $\|\boldsymbol{\delta}\|_2 \leq g(n)$ where $g(n) \to 0$ for $n \to \infty$. Further, we also want to prove non-asymptotic guarantees for statistical recovery.

We study the behavior of high dimensional estimators in the presence of noise and present three key findings:

- First, we exploit the current bounding techniques [14, 12] and show that the error of regularized estimators in the presence of noise based on current techniques can only be bounded by two terms one of which shrinks as the number of samples increases and the other one is irreducible and depends on the covariance of the noise.

- Second, when an estimate of the noise covariance is known, we show that existing estimators [12, 9] provide consistent estimates for any norm regularization $R(\cdot)$. Our analysis generalizes the existing estimators in the noisy setting, which have only considered sparse regression and l_1 norm regularization.

- Finally, using LASSO as the estimator, we empirically show that in the presence of noise in covariates, even estimation followed by significant test fails to detect all important features, whereas our estimator, having knowledge of noise covariance, captures relevant features more accurately.

1.3.3 Dictionary Learning for Sentiment Analysis

Twitter, a micro-blogging website, is among the most pervasive social media platforms. On a regular basis, it's users *willingly* share their thoughts, preferences, and emotions, in the form of messages 140 characters in length (a.k.a. tweets). Although the field of social media analytics for a rich source of information, like weblogs, is becoming mature, the microblog analysis is in its early stages of life.

Several data mining tasks can be defined for Twitter data with various applications. Among them sentiment analysis [45, 46] has increasingly gained attentions from both academia and industry. The post length constraint causes the feature space of tweets to be very sparse, which renders determining the positive or negative sense of a tweet difficult even for a human judge.

Our contribution in this work is threefold:

- To the best of our knowledge, we are the first to present a complete pipeline for Twitter sentiment analysis.
- We introduce weighted dictionary learning for classification of uncertain-labeled tweets.
- We empirically show that sparsity of tweets enables us to perform classification with their low dimensional random projections without losing accuracy.

1.3.4 Influence Maximization in Social Networks

In this manuscript, we propose and develop a powerful *heat conduction* (HC) framework for modeling and studying the influence maximization problem under the *non-progressive* influence process, where an *activated* node can be reverted to *inactive* subsequently. The non-progressive influence diffusion process more realistically captures a wide variety of real-life applications and scenarios where users' opinions, interests, and behaviors can change over time when exposed to different sources of influence. The HC framework unifies, generalizes, and extends the existing non-progressive models. Our contribution in this work is summarized as follows:

- We propose HC influence model that has favorable real world interpretations and unifies, generalizes, and extends the existing non-progressive models.
- We show HC has three distinctive key properties which enable us solving influence maximization (1.10) efficiently.
- We demonstrate high performance and scalability of our algorithm via extensive experiments and present the first real non-progressive cascade dataset.

1.4 Notations and Preliminaries

Notation. We denote sets with curly characters $\mathcal{V}$, matrices by bold capital letters $\mathbf{V}$, random variables by capital letters V, vectors by small bold symbols $\mathbf{v}$ which are indexed with either a single number as $\mathbf{v}(i)$ or an index set $\mathcal{A}$ as $\mathbf{v}_{\mathcal{A}}$. Row i of the matrix $\mathbf{V}$ is shown as $\mathbf{v}_i$ and jth element of the vector $\mathbf{v}$ is shown as $\mathbf{v}(j)$. The (i, j)th element of the matrix $\mathbf{V}$ is shown in three ways: $\mathbf{V}_{ij}$, $\mathbf{v}_i(j)$, or v_{ij}. Throughout the manuscript c_i and C_i are positive constants.

Sub-Gaussian (Sub-exponential) random variable and vector. A random variable V is sub-Gaussian (sub-exponential) if the moments satisfies

$$\forall p \geq 1 : (\mathbb{E}|V|^p)^{1/p} \leq K_2\sqrt{p} \left((\mathbb{E}|V|^p)^{1/p} \leq K_2 p\right) \tag{1.16}$$

The minimum value of K_2 (K_1) is called sub-Gaussian (sub-exponential) norm of V, denoted by $|||V|||_{\psi_2}(|||V|||_{\psi_1})$ [47]. A random vector $\mathbf{v} \in \mathbb{R}^p$ sub-Gaussian (sub-exponential) if the one-dimensional marginals $\langle \mathbf{v}, \mathbf{u} \rangle$ are sub-Gaussian (sub-exponential) random variables for all $\mathbf{u} \in \mathbb{R}^p$. The sub-Gaussian (sub-exponential) norm of $\mathbf{v}$ is defined [47]:

$$|||\mathbf{v}|||_{\psi_2} = \sup_{\mathbf{u} \in \mathbb{S}^{p-1}} |||\langle \mathbf{v}, \mathbf{u} \rangle|||_{\psi_2}, \quad (|||\mathbf{v}|||_{\psi_1} = \sup_{\mathbf{u} \in \mathbb{S}^{p-1}} |||\langle \mathbf{v}, \mathbf{u} \rangle|||_{\psi_1}) \tag{1.17}$$

We abuse notation and use shorthand $\mathbf{v} \sim \text{Subg}(0, \Sigma_{\mathbf{v}}, K_{\mathbf{v}})$ for zero mean sub-Gaussian random vector with covariance $\Sigma_{\mathbf{v}}$ and sub-Gaussian norm of $K_{\mathbf{v}}$, although keeping in mind that no other moments, nor the exact form of the distribution function is known. For any set $\mathcal{V} \in \mathbb{R}^p$ the Gaussian width of the set $\mathcal{V}$ is defined as $\omega(\mathcal{V}) = \mathbb{E}_{\mathbf{g}}\left[\sup_{\mathbf{u} \in \mathcal{V}} \langle \mathbf{g}, \mathbf{u} \rangle\right]$ [42], where the expectation is over $\mathbf{g} \sim N(\mathbf{0}, \mathbf{I}_{p \times p})$, a vector of independent zero-mean unit-variance Gaussian.

We define the minimum and maximum eigenvalues of a matrix $\mathbf{M}$ restricted to the set $\mathcal{A} \subseteq \mathbb{S}^{p-1}$ as $\lambda_{\min}(\mathbf{M}|\mathcal{A}) = \inf_{\mathbf{u} \in \mathcal{A}} \mathbf{u}^T \mathbf{M} \mathbf{u}$, and $\lambda_{\max}(\mathbf{M}|\mathcal{A}) = \sup_{\mathbf{u} \in \mathcal{A}} \mathbf{u}^T \mathbf{M} \mathbf{u}$ respectively.

All c_i, c, and C represent universal constants throughout the manuscript. Set $[G] = \{0, \dots, G\}$ is the index set for both shared and private components (in the setting of data sharing model (1.4)) and $[G]_\setminus = [G] - \{0\}$ represents only the private ones.

Chapter 2

Related Work

In this chapter, we briefly review the relevant literature of the four problems, Table 1.1, that we are studying in this manuscript.

2.1 Data Sharing Model

The high dimensional data sharing model has recently gained attention because of its wide range of application such as personalized medicine [26, 21], sentiment analysis and banking strategy [21], single cell data analysis [28], road safety [27], and disease subtype analysis [26]. More generally, in any high dimensional domain where the population consists of groups or clusters, the data sharing framework has the potential to boost both perdition and parameter recovery.

In spite of the recent surge in applying the data sharing framework to different domains, there is little known about statistical properties of the proposed estimators. In fact, non-asymptotic statistical properties of the regularized estimator for the data sharing model is still an open question [21, 27]. To the best of our knowledge, the only theoretical guarantee for data sharing is provided in [28] where the authors, under the stringent irrepresentability condition of the design matrix, prove sparsistency of their proposed method. Beyond sparsity and l_1-norm, no other structure has been investigated for these models.

Like any high dimensional model, questions about the data sharing model concerns sample complexity required for recovering the parameters and the non-asymptotic rate of estimation error. Here, the more interesting question is about the shared parameter. Does the sample complexity of $\boldsymbol{\beta}_0^*$ depend on the data of all groups? What about the rate of the error bound? In other words, we are investigating the conjecture that data pooled from all groups will facilitate estimation of the shared parameter in regards of sample complexity and error rate. In this work, we explicitly answer these questions.

2.2 Error-in-Variable Models

Over the past decade considerable progress has been made on the sparse and structured estimation problems for linear models. Such models assume that the observed pair $(\mathbf{x}_i, y_i)$ follow $y_i = \langle \boldsymbol{\beta}^*, \mathbf{x}_i \rangle + \epsilon_i$, where $\boldsymbol{\beta}^*$ is sparse or suitably structured according to a norm R [48, 14, 42, 12]. In real world settings, often covariates are noisy, and one observes $\mathbf{x}_i$ corrupted by noise $\mathbf{w}_i$, generally stated as $\mathbf{z}_i = f(\mathbf{x}_i, \mathbf{w}_i)$. Two popular model for f are additive, $\mathbf{z}_i = \mathbf{x}_i + \mathbf{w}_i$, and multiplicative noise $\mathbf{z}_i = \mathbf{x}_i \circ \mathbf{w}_i$ [49, 22, 50] where $\circ$ is the Hadamard product. Two common noise models for $\mathbf{w}_i$ are uniformly bounded [51, 50] and centered sub-Gaussian [49, 22]. In noisy models, a key challenge is to develop estimation methods that are robust to corrupted data, particularly in the high-dimensional regime. Recent work [49, 50] has illustrated empirically that standard estimators like LASSO and Dantzig Selector (DS) [11] perform poorly in the presence of measurement errors. Thus, many recent papers proposed modifications to LASSO, DS or Orthogonal Matching Pursuit (OMP) [51, 49, 22, 50, 52] for handling noisy covariates. However, such estimators may become non-convex [22], or require extra information about optimal $\boldsymbol{\beta}^*$ [49, 22]. Further, most of proposed estimators for sub-Gaussian additive noise require an estimate of the noise covariance $\Sigma_{\mathbf{w}}$ in order to establish statistical consistency [51, 49, 22, 52] or impose more stringent condition, like element-wise boundedness on $\mathbf{W}$, the random noise matrix [51, 50].

Name	Estimator	Conditions	Bound for $\lVert\Delta\rVert_2$
MU [50]	min $\lVert\boldsymbol{\beta}\rVert_1$ s.t. $\lVert\frac{1}{n}Z^T(\mathbf{y}-Z\boldsymbol{\beta})\rVert_\infty$ $\le(1+\delta)\delta\lVert\boldsymbol{\beta}\rVert_1+\tau$	$\lVert\frac{1}{n}Z^T\boldsymbol{\epsilon}\rVert_\infty\le\tau$ $\forall W_{ij},\lvert W_{ij}\rvert\le\delta$	$c\sqrt{s}(\delta+\delta^2)\lVert\boldsymbol{\beta}^*\rVert_1$ $+C\sqrt{\frac{s\log p}{n}}$
IMU [52]	min $\lVert\boldsymbol{\beta}\rVert_1$ s.t. $\lVert\frac{1}{n}Z^T(\mathbf{y}-Z\boldsymbol{\beta})+\hat{\Sigma}_{\mathbf{w}}\boldsymbol{\beta}\rVert_\infty$ $\le\mu\lVert\boldsymbol{\beta}\rVert_1+\tau$	$\sigma_j^2=\frac{1}{n}\sum_{i=1}^n\mathbb{E}W_{ij}^2$ $\Sigma_{\mathbf{w}}=$ diag$(\sigma_1,\ldots,\sigma_p)$ $\mathbf{w}_i\sim$ Subg$(0,\Sigma_{\mathbf{w}},K_{\mathbf{w}})$	$C\lVert\boldsymbol{\beta}^*\rVert_1\sqrt{\frac{s\log p}{n}}$
NCL [22]	min $\frac{1}{2}\boldsymbol{\beta}^T(\frac{1}{n}Z^TZ-\Sigma_{\mathbf{w}})\boldsymbol{\beta}$ $-\frac{1}{n}\boldsymbol{\beta}^TZ^T\mathbf{y}+\lambda\lVert\boldsymbol{\beta}\rVert_1$ s.t. $\lVert\boldsymbol{\beta}\rVert_1\le b_1$	$\mathbf{w}_i\sim$ Subg$(0,\Sigma_{\mathbf{w}},K_{\mathbf{w}})$	$\max\{c\sqrt{s}\lambda,C\lVert\boldsymbol{\beta}^*\rVert_2\sqrt{\frac{s\log p}{n}}\}$
NCC [22]	min $\frac{1}{2}\boldsymbol{\beta}^T(\frac{1}{n}Z^TZ-\Sigma_{\mathbf{w}})\boldsymbol{\beta}$ $-\frac{1}{n}\boldsymbol{\beta}^TZ^T\mathbf{y}$ s.t. $\lVert\boldsymbol{\beta}\rVert_1\le b_2$	$\mathbf{w}_i\sim$ Subg$(0,\Sigma_{\mathbf{w}},K_{\mathbf{w}})$	$C\lVert\boldsymbol{\beta}^*\rVert_2\sqrt{\frac{s\log p}{n}}$
OMP [49]	OMP to recover indecies S: $\hat{\boldsymbol{\beta}}_S=(Z_S^TZ_S-\Sigma_{\mathbf{w}}^S)(Z_S^T\mathbf{y})$	$\mathbf{w}_i\sim$ Subg$(0,\Sigma_{\mathbf{w}},K_{\mathbf{w}})$ $\forall\beta_i^*\neq0$ $\lvert\beta_i^*\rvert\ge(c\lVert\boldsymbol{\beta}\rVert_2+C)\sqrt{\frac{\log p}{n}}$	$(c+C\lVert\boldsymbol{\beta}^*\rVert_2)\sqrt{\frac{s\log p}{n}}$

Table 2.1: Comparison of estimators for design corrupted with additive sub-Gaussian noise.

Table 2.1 presents key recent literature on regression with additive measurement error in high dimension focusing on sparsity. The first paper in this line of work [50] introduces matrix uncertainty selector (MU) which belongs to constraint family of estimators. As the first attempt for addressing estimation with measurement error in high dimension, MU imposes restrictive conditions on noise W, namely each element of matrix W needs to be bounded. It worth mentioning that MU does not need any information about the noise covariance $\Sigma_{\mathbf{w}}$ but as presented in Table 2.1, it is not consistent. The term $c\sqrt{s}(\delta+\delta^2)\|\boldsymbol{\beta}^*\|_1$ in the upper bound is independent of the number of samples n. This theme repeats in the literature: when $\Sigma_{\mathbf{w}}$ is available proposed estimators are consistent otherwise there is no l_2 recovery guarantee.

The improved matrix uncertainty selector (IMU) [52] assumes the availability of the diagonal matrix $\hat{\Sigma}_{\mathbf{w}}$ as the covariance of the noise and uses it to compensate the effect of the noise. The compensation idea also recurs in the literature where one mitigates Z^TZ by subtracting $\Sigma_{\mathbf{w}}$, and as a result the estimator becomes consistent. Note that both MU and IMU are variants of DS where $\|\boldsymbol{\beta}\|_1$ appears in both constraint and objective of the optimization program. For IMU each row of the noise matrix $\mathbf{w}_i$ is sub-Gaussian and independent of $\mathbf{w}_j$, $\mathbf{x}_i$ and ϵ_i. Also, the off diagonals of $\Sigma_{\mathbf{w}}$ are zero meaning W_{ij} are uncorrelated. Following IMU all subsequent work assume sub-Gaussian independent noise. The MU and [51] are only estimators that allow general dependence in noise.

Loh and Wainwright [22] proposed a non-convex modification of LASSO (NCL) [9] along with constraint version of it (NCC) which are equivalent by Lagrangian duality (Table 2.1). In both estimators, they substitute the quadratic term $X^T X$ of the LASSO objective with $Z^T Z - \Sigma_{\mathbf{w}}$ which makes the problem non-convex. An interesting aspect of this method is that although a projected gradient algorithm can only reach a local minimum, yet any such local minima is guaranteed to have consistency guarantee. Note that for the feasibility of both objectives, [22] requires extra information about the unknown parameter $\boldsymbol{\beta}^*$, particularly b_1 and b_2 should be set to a value greater than $\|\boldsymbol{\beta}^*\|_1$.

In [49], Chen and Caramanis use the OMP [53] for support recovery of a sparse regression problem without knowing the noise covariance. They established non-asymptotic guarantees for support recovery while imposing element-wise lower bound on the absolute value of the support. However, for achieving l_2 consistently, [49] still requires an estimate of the noise covariance $\Sigma_{\mathbf{w}}$, which is in accordance with the requirements of other estimators mentioned above.

Although the literature on regression with noisy covariates has only focused on sparsity, the machine learning community recently has made tremendous progress on the structured regression, i.e., beyond l_1-norm, that has led to several key publications. The work [12] provided a general framework for analyzing regularized estimators with decomposable norm of the form $\min_{\boldsymbol{\beta}} \mathcal{L}(\boldsymbol{\beta}; \mathbf{y}, X) + \lambda R(\boldsymbol{\beta})$, and established theoretical guarantees for Gaussian covariates. Recent papers [54, 55] have generalized this framework for analyzing estimators with hierarchical structures [56], atomic norms [54] and graphical model structure learning [55]. Lately, [14] established a framework for analyzing regularized estimators with any norm $R(\cdot)$ and sub-Gaussian covariates. For constraint estimators, [57] has recently generalized the DS for any norm $R(\cdot)$. Building upon these advances, we extend the literature of high-dimensional regression with noisy covariates beyond l_1-norm.

2.3 Dictionary Learning

Sparse modeling techniques have gained popularity among the signal processing and machine learning communities for their ability to provide efficient representations of a great variety of signals such as audio and natural images. This efficiency is achieved by approximating a signal with a linear combination of a few elements (atoms) of some (often) redundant bases. When these bases are learned from the data itself, they are called dictionaries [40].

Formally, we aim at learning a dictionary $\mathbf{D} \in \mathbb{R}^{m \times k}$ such that a training set of signals $\mathcal{X} = \{\mathbf{x}_i \in \mathbb{R}^m \mid i = 1, \ldots, n\}$ (and later testing data from the same class) can be well represented by linearly combining a few of the basis vectors formed by the columns of $\mathbf{D}$. This problem can be cast as the optimization

$$\min_{\substack{\mathbf{D}, \boldsymbol{\alpha}_i \\ i=1,\ldots,n}} \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2} \|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda \|\boldsymbol{\alpha}_i\|_1, \tag{2.1}$$

which is convex with respect to the variables $\boldsymbol{\alpha}_i$ when $\mathbf{D}$ is fixed and viceversa (here λ a positive constant). The optimization is then commonly solved by alternatively fixing one and minimizing over the other.

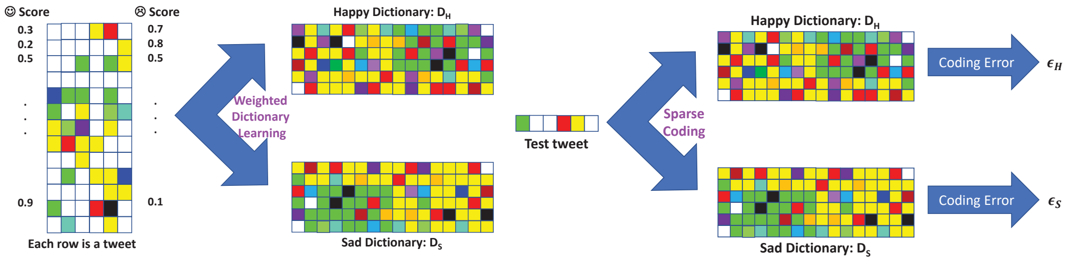
Sparse modeling has been previously employed for supervised classification tasks, exhibiting state-of-the-art performance in visual and audio applications such as face recognition and the PASCAL challenge [58].

Classification is often done by learning, following the above optimization, a dictionary $\mathbf{D}_c$ for each class $c \in \mathcal{C}$ using only training data from the set $\{\mathbf{x}_i \in \mathcal{X} \mid y_i = c\}$. Classification is then performed with testing data $\mathcal{X}_{\text{test}}$, assigning a label $c^* = f(\mathbf{x})$ to each $\mathbf{x} \in \mathcal{X}_{\text{test}}$ where

$$f(\mathbf{x}) = \operatorname*{argmin}_{c \in \mathcal{C}} \ \ell(\mathbf{x}, \mathbf{D}_c)$$

where $\ell(\mathbf{x}, \mathbf{D}_c) = \min_{\alpha} \frac{1}{2} \|\mathbf{x} - \mathbf{D}_c \boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1$.

In this work we generalize the dictionary learning problem (2.1) for the case of weighted input data and present an algorithm for solving the *weighted dictionary learning* problem. The proposed algorithm is then applied to the tweet sentiment analysis problem as discussed in Section (1.1.1). For example, using the probabilistically labeled tweets we learn two dictionaries representing “happy” and “sad” moods, Figure 2.1a, and use them to determine the label of the new tweets, 2.1b.



(a) Learning dictionary for sentiments.

(b) Labeling new tweet.

Figure 2.1: Weighted dictionary learning for classification [1].

2.4 Influence Maximization

Most influence maximization studies have focused on the *progressive* influence processes where once a node is activated, it cannot be reverted. Figure 2.2 shows the progressive process of buying a new cell phone. When someone buys the new product the increase in the revenue progresses and the new buyer may influence others to purchase the product too.

In the seminal work [24], Kempe *et al* show that the *progressive* influence maximization problem under both the linear threshold (LT) and independent cascade (IC) diffusion models is NP-hard. On the other hand, the progressive influence maximization problem can be well approximated by establishing that the influence function is submodular. In practice, however, solving the progressive influence maximization problem is still computationally expensive for large social networks, due to the need for estimating the influence function $\sigma(\mathcal{S})$ which has no known closed-form and is estimated by Monte Carlo simulation. As mentioned in Section 1.3.4, the choice of influence model is important because it determines the computational hardness of computing and operating with the influence function $\sigma(\mathcal{S})$. In other words, although the oracle[1] complexities of the submodularity based algorithms are reasonable, since computation of $\sigma(\mathcal{S})$ is expensive, they are not scalable for influence maximization.

The follow-up studies [59][60][61][62][63][64] attempt to speed up this process by avoiding or decreasing the need for the Monte Carlo simulation. The CELF method of Leskovec et al. [59] attempts to speed up the original greedy method, proposed by Kempe et al. [24], by reducing the number of calls to the Monte Carlo routine for spread computation. The CELF lazy method is based on the submodularity of the influence spread and can be applied to any

[1] Where given $\mathcal{S}$ the oracle provides us with function value $\sigma(\mathcal{S})$

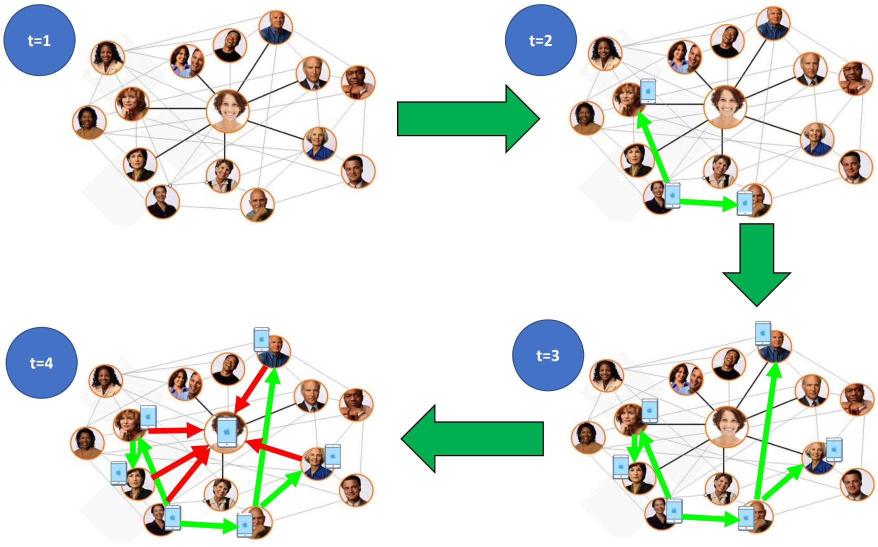


Figure 2.2: An example of progressive influence model.

submodular maximization problem. Although lazy evaluation improves the running time of the original greedy method by up to 700 times [59], it still does not scale to large graphs [60].

Recently heuristics have been proposed to approximate influence spread for LT [60] and IC [61] which enables the greedy method to scale for large networks. Chen et al. [60] suggest using a local directed acyclic graph (LDAG) per node, instead of considering the whole graph, to approximate the influence flowing to the node. Goyal et al. propose SIMPATH method [62] under the LT model which is built on the CELF method [59]. They approximate the influence spread by enumerating the simple paths starting from the seeds within a small neighborhood. Both of these methods have parameters to be tuned which control the trade-off between running time and accuracy of influence spread estimation. Methods presented in [60, 62] accelerate the greedy method [24] substantially and achieve high performance in influence maximization.

Gomez-Rodriguez et al. [63] propose a progressive continuous time influence model with dynamics similar to IC and show that influence maximization is NP-hard for this model as well. They show submodularity of influence spread and exploit the same greedy algorithm. In contrast to all other progressive models, influence spread has a closed form for this model, but the computation is not scalable for large scale networks. A recent work [64] has scaled influence computation by developing a randomized algorithm for approximating it.

Beyond progressive influence models, little work has been done on non-progressive models.

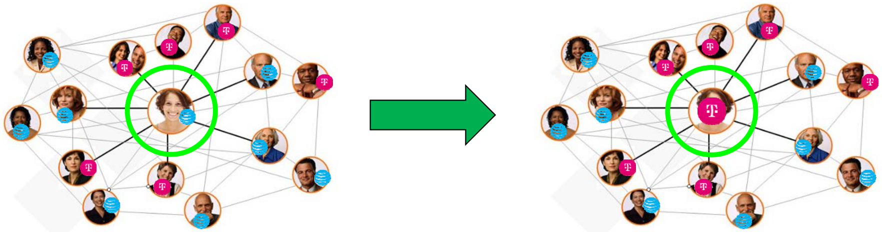


Figure 2.3: Switching between carriers makes the revenue of companies non-progressive.

Non-progressive models are better at modeling market shares of different products and capturing the spread of the products where customer's loyalty is the source of revenue. For example, users can switch between cell-phone carriers at any time, Figure 2.3 which changes the market share of each carrier and the overall loyalty of the customers determines the total revenue.

Kempe et al. [24] introduce a non-progressive version of the LT influence model (NLT) and try to tackle the influence maximization problem under NLT by reducing the model to (progressive) LT, discussed in Chapter 6. Voter model, as the most well-known non-progressive model, is originally introduced in [65, 66] and adopted for viral marketing in [67]. Even-Dar and Shapira show that under Voter model, highest degree nodes are the solution of influence maximization [67]. Unfortunately, since the Voter model reaches consensus, i.e. one product remains in the long term, it can not explain the coexistence of multiple product adoptions, which is a typical case in many real product adoptions.

Kempe et al. [24] try to tackle the influence maximization problem under NLT by reducing the NLT model to (progressive) LT. For this purpose, they replicate the social network for each time step where each node has a copy in each time and connects to its neighbors in the previous copy of the network. This trick reduces the non-progressive model to a progressive one but obviously increases the computational complexity and clearly does not work for infinite time horizon. Here, we propose a powerful non-progressive influence model named *heat conduction* (HC) model, and study the influence maximization problem for it. We show that HC model unifies, generalizes, and extends the existing non-progressive models, specifically we generalize the NLT and present a scalable algorithm to solve the influence maximization under NLT.

Part II

Continuous Problems

Chapter 3

Structured High Dimensional Data Sharing Model

The high dimensional structured data sharing model describes groups of observations by shared and per-group private parameters, each with its own structure such as sparsity or group sparsity. In this chapter we consider the general form of data sharing where data comes in a fixed but arbitrary number of groups G and the structure of both shared and private parameters can be characterized by any norm. We propose a simple estimator for the high dimensional data sharing model and provide conditions under which it consistently estimates both shared and private parameters. We also characterize sample complexity of the estimator and present high probability non-asymptotic bounds on estimation errors of all parameters. Interestingly the sample complexity of our estimator translates to conditions on both per-group sample size and total number of samples. To the best of our knowledge, this is the first thorough statistical analysis of data sharing models. This is important because of its recent wide spread.

The rest of this chapter is organized as follows: We start with the details of the problem setup and introduce our estimator in Section 3.1. In Section 3.2, we characterize the error set of our estimator and provide a deterministic error bound. In Section 3.3 we discuss the restricted eigenvalue condition and calculate the per-group and total sample complexity required for recovery of the true parameters by our estimator. In Section 3.4 we close the statistical analysis by providing high probability error bounds. Finally, we provide a linearly convergent algorithm for finding the solution of our estimator in Section 3.5. We present experimental results on Section 3.6. The proofs of all technical results are detailed in Section 3.7.

3.1 Problem Setup and The Estimator

Given G group and n_g samples in each one as $\{\{\mathbf{x}_{gi}, y_{gi}\}_{i=1}^{n_g}\}_{g=1}^{G}$, we can form the per group design matrix $\mathbf{X}_g \in \mathbb{R}^{n_g \times p}$ and output vector $\mathbf{y}_g \in \mathbb{R}^{n_g}$. The total number of samples is $n = \sum_{g=1}^{G} n_g$. The data sharing model takes the following vector form:

$$\mathbf{y}_g = \mathbf{X}_g(\boldsymbol{\beta}_0^* + \boldsymbol{\beta}_g^*) + \boldsymbol{\omega}_g, \quad \forall g \in [G]_{\setminus} \tag{3.1}$$

where each row of $\mathbf{X}_g$ is $\mathbf{x}_{gi}^T$ and $\boldsymbol{\omega}_g^T = (\omega_{g1}, \dots, \omega_{gn_g})$ consists of i.i.d. centered unit-variance sub-Gaussian elements with $|||\omega_{gi}|||_{\psi_2} \leq K$. The shared parameter among all groups is $\boldsymbol{\beta}_0^*$ and the private parameter of the group g is $\boldsymbol{\beta}_g^*$. We focus on independent isotropic sub-Gaussian random vectors $\mathbf{x}_{gi}$ where $|||\mathbf{x}_{gi}|||_{\psi_2} \leq k$ and $\mathbb{E}\mathbf{x}_{gi}^T\mathbf{x}_{gi} = \mathbf{I}_{p\times p}$. Extension to anisotropic sub-Gaussian case is straightforward by techniques developed in the recent literature [14, 68].

For shared and private parameters recovery, we propose the following estimator :

$$\hat{\boldsymbol{\beta}} = (\hat{\boldsymbol{\beta}}_0^T, \dots, \hat{\boldsymbol{\beta}}_G^T) \in \operatorname*{argmin}_{\boldsymbol{\beta}_0,\dots,\boldsymbol{\beta}_G} \frac{1}{n} \sum_{g=1}^{G} \|\mathbf{y}_g - \mathbf{X}_g(\boldsymbol{\beta}_0 + \boldsymbol{\beta}_g)\|_2^2, \quad \forall g \in [G] : R_g(\boldsymbol{\beta}_g) \leq R_g(\boldsymbol{\beta}_g^*) \tag{3.2}$$

Interestingly, we can write a compact optimization problem that is equivalent to (3.2) as:

$$\hat{\boldsymbol{\beta}} \in \operatorname*{argmin}_{\boldsymbol{\beta}} \frac{1}{n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2, \quad \forall g \in [G] : R_g(\boldsymbol{\beta}) \leq R_g(\boldsymbol{\beta}^*), \tag{3.3}$$

where $\mathbf{y}^T = (\mathbf{y}_1^T, \dots \mathbf{y}_G^T) \in \mathbb{R}^n$, $\boldsymbol{\beta} = (\boldsymbol{\beta}_0{}^T, \dots, \boldsymbol{\beta}_G{}^T)^T \in \mathbb{R}^{(G+1)p}$ and

$$\mathbf{X} = \begin{pmatrix} \mathbf{X}_1 & \mathbf{X}_1 & 0 & \cdots & 0 \\ \mathbf{X}_2 & 0 & \mathbf{X}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \cdots & \vdots \\ \mathbf{X}_G & 0 & \cdots & \cdots & \mathbf{X}_G \end{pmatrix} \in \mathbb{R}^{n\times(G+1)p} \tag{3.4}$$

For simplicity we denote $\mathbf{X} = [\mathbf{W} \;\; \mathbf{D}]$ which is the concatenation of $\mathbf{W} \in \mathbb{R}^{n\times p}$ that represents the *whole* design matrix consists of all data points as rows and $\mathbf{D} \in \mathbb{R}^{n\times pG}$ which is the *diagonal* part of the $\mathbf{X}$ where all $\mathbf{X}_g$s are on the diagonal.

3.2 Error Set and Deterministic Error Bound

Since $\hat{\boldsymbol{\beta}}_g = \boldsymbol{\beta}^*_g + \boldsymbol{\delta}_g$ is a feasible point of the optimization (3.2), $\boldsymbol{\delta}_g$ will blelong to the following restricted error set, which is the set of all descent directions at $\boldsymbol{\beta}^*_g$ on $R_g(\cdot)$:

$$\mathcal{E}_g = \left\{\boldsymbol{\delta}_g | R(\boldsymbol{\beta}^*_g + \boldsymbol{\delta}_g) \leq R(\boldsymbol{\beta}^*_g)\right\}, \quad g \in [G]$$

We name the cone of the error set as $\mathcal{C}_g = \text{Cone}(\mathcal{E}_g)$ and the spherical cap corresponding to it as $\mathcal{A}_g = \mathcal{C}_g \cap \mathbb{S}^{p-1}$. Subsequently, we define the following set:

$$\mathcal{H} \quad = \quad \left\{ \boldsymbol{\delta} = (\boldsymbol{\delta}_0^T, \dots, \boldsymbol{\delta}_G^T)^T \middle| \forall g \in [G] : \boldsymbol{\delta}_g \in \mathcal{C}_g, \sum_{g=0}^{G} \alpha_g \|\boldsymbol{\delta}_g\|_2 = 1 \right\}, \alpha_g = \sqrt{\frac{n_g}{n}}.$$

Starting from the optimality of $\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}^* + \boldsymbol{\delta}$ as $\frac{1}{n}\|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}\|_2^2 \leq \frac{1}{n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^*\|_2^2$ we derive:

$$\frac{1}{n}\|\mathbf{X}\boldsymbol{\delta}\|_2^2 \quad \leq \quad \frac{1}{n} 2\boldsymbol{\omega}^T \mathbf{X}\boldsymbol{\delta} \tag{3.5}$$

where $\boldsymbol{\omega} = [\boldsymbol{\omega}_1^T, \dots, \boldsymbol{\omega}_G^T]^T \in \mathbb{R}^n$ is the vector of all noises.

Using the basic inequality (3.5) we can establish the following deterministic error bound.

Theorem 3.1 *For the estimator proposed in* (3.3)*, assume that there exist* $0 < \kappa \leq \inf_{\mathbf{u} \in \mathcal{H}} \frac{1}{n}\|\mathbf{X}\mathbf{u}\|_2^2$. *Then we have the following upper bound for weighted sum of error:*

$$\sum_{g=0}^{G} \alpha_g \|\boldsymbol{\delta}_g\|_2 \leq \frac{2 \sup_{\mathbf{u} \in \mathcal{H}} \boldsymbol{\omega}^T \mathbf{X}\mathbf{u}}{n\kappa}, \quad \alpha_g = \sqrt{\frac{n_g}{n}}.$$

3.3 Restricted Eigenvalue Condition for Data Sharing Model

The main assumption of Theorem 3.1 is known as Restricted Eigenvalue (RE) condition in the literature of high dimensional statistics [14, 12, 69]:

$$\inf_{\mathbf{u} \in \mathcal{H}} \frac{1}{n}\|\mathbf{X}\mathbf{u}\|_2^2 \geq \kappa > 0 \tag{3.6}$$

The RE condition (3.6) assumes that the minimum eigenvalues of the matrix $\mathbf{X}^T\mathbf{X}$ in directions restricted to $\mathcal{H}$ is strictly positive. In this section, we want to show that for the data sharing design matrix $\mathbf{X}$ defined in (3.4), the RE condition (3.6) holds with high probability when we have enough number of samples, known as sample complexity.

Note that each of the linear models of (3.1) is a superposition [70] or dirty statistical model [71]. Therefore, we have a set of coupled superposition models, and the goal is to estimate their parameters. Another similar model is the one used in [72], but the authors are not emphasizing on a distinct shared component. The straightforward way to get the sample complexity for satisfying the RE condition is to use results from the superposition literature directly. Here we focus on the state-of-the-art estimator proposed in [70], and show it will not lead to a reasonable sample complexity.

Proposition 1 *Using the RE condition analysis of superposition model of [70], recovering the shared parameter $\boldsymbol{\beta}_0^*$ requires at least one group to have $n_g \geq \omega^2(\mathcal{A}_0)$. To recover each private parameter we also need at least $n_g \geq \max(\omega(\mathcal{A}_0)^2, \omega(\mathcal{A}_g)^2)$ samples in the group. In other words, by separate analysis of superposition estimators neither the recovery of shared parameter benefits from the pooled n samples, nor the private parameters.*

Proof: Note that $\mathbf{y}_g = \mathbf{X}_g(\boldsymbol{\beta}_g^* + \boldsymbol{\beta}_0^*) + \boldsymbol{\omega}_g$ is a superposition model and as shown in [70] the sample complexity required for the RE condition and subsequently recovering $\boldsymbol{\beta}_0^*$ and $\boldsymbol{\beta}_g^*$ is $n_g \geq c(\max(\omega(\mathcal{A}_0), \omega(\mathcal{A}_g)) + \sqrt{\log 2})^2$.

Note that Proposition 1 suggests that the sample complexity for the recovery of the shared parameter and private parameters are coupled together based on the current state-of-the-art analysis while one hopes to get a decoupled version of them. To improve this sample complexity and exploit all data for recovering the shared parameter, the key is to use the block decomposition of the design matrix as $\mathbf{X} = [\mathbf{W} \;\; \mathbf{D}]$. Using this decompostion the RE condition becomes:

$$\inf_{\boldsymbol{\delta} \in \mathcal{H}} \|\mathbf{X}\boldsymbol{\delta}\|_2^2 = \inf_{\boldsymbol{\delta} \in \mathcal{H}} \|\mathbf{W}\boldsymbol{\delta}_0 + \mathbf{D}\boldsymbol{\delta}_{1:G}\|_2^2,$$

where $\boldsymbol{\delta}_{1:G} = [\boldsymbol{\delta}_1^T, \ldots, \boldsymbol{\delta}_G^T]^T$. We want to reduce the RE condition on set $\mathcal{H}$ with design $\mathbf{X}$, to the RE conditions of $\boldsymbol{\delta}_0 \in \mathcal{C}_0$ and $\boldsymbol{\delta}_{1:G}$ with designs $\mathbf{W}$ and $\mathbf{D}$ respectively. In the following, we elaborate this decoupling step. The below lemma is a reverse of triangle inequality which plays a key role in our decoupling step.

Lemma 3.2 (Proposition A.2. of [73]) *If there exists $\epsilon \in (0, 1]$ such that $-\langle x, y\rangle \leq (1 - \epsilon)\|x\|_2\|y\|_2$, then:*

$$\|x + y\|_2^2 \geq \epsilon(\|x\|_2^2 + \|y\|_2^2) \tag{3.7}$$

Next lemma establishes the RE condition for individual isotropic sub-Gaussian designs and provides us with the essential tool for proving high probability bounds.

Lemma 3.3 (Theorem 11 of [14]) *To unify the illustration assume,* $n_0 = n$ *and* $\mathbf{X}_0 = \mathbf{W}$*. For all* $g \in [G]$*, for the matrix* $\mathbf{X}_g \in \mathbb{R}^{n_g \times p}$ *with independent isotropic sub-Gaussian rows, i.e.,* $|||\mathbf{x}_{gi}|||_{\psi_2} \leq k$ *and* $\mathbb{E}[\mathbf{x}_{gi}\mathbf{x}_{gi}^T] = \mathbf{I}$*, following results hold on the spherical cap* $\mathcal{A}_g = \mathcal{C}_g \cap \mathbb{S}^{p-1}$ *with probability at least* $1 - 2\exp\left(-\gamma_g(\omega(\mathcal{A}_g) + \tau)^2\right)$ *for* $\tau > 0$*:*

$$\begin{aligned} n_g - c_g\sqrt{n_g}\omega(\mathcal{A}_g) - C_g\sqrt{n_g}\tau &\leq \inf_{\mathbf{u}\in\mathcal{A}_g} \|\mathbf{X}_g\mathbf{u}\|_2^2 \\ &\leq \sup_{\mathbf{u}\in\mathcal{A}_g} \|\mathbf{X}_g\mathbf{u}\|_2^2 \leq n_g + c_g\sqrt{n_g}\omega(\mathcal{A}_g) + C_g\sqrt{n_g}\tau \end{aligned}$$

where $c_g, C_g > 0$ *are constants.*

The statement of Lemma 3.3 characterizes the distortion in the Euclidean distance between points $\mathbf{u}_g \in \mathcal{C}_g$ when the matrix $\mathbf{X}_g$ is applied to them and states that any sub-Gaussian design matrix is approximately isometry, with high probability:

$$(1-\alpha)\|\mathbf{u}_g\|_2 \leq \|\mathbf{X}_g\mathbf{u}_g\|_2^2 \leq (1+\alpha)\|\mathbf{u}_g\|_2$$

where $\alpha = c_g\frac{\omega(\mathcal{A}_g)}{\sqrt{n_g}}$.

Using the result of Lemma 3.3, in the following lemma, we show that the assumption of the Lemma 3.2 holds for the two n-dimensional vectors $\mathbf{W}\boldsymbol{\delta}_0$ and $\mathbf{D}\boldsymbol{\delta}_{1:G}$ with high probability.

Lemma 3.4 *For* $\mathbf{W}\boldsymbol{\delta}_0 \in \mathbb{R}^n$ *and* $\mathbf{D}\boldsymbol{\delta}_{1:G} \in \mathbb{R}^n$ *where* $\mathbf{W}$ *and* $\mathbf{D}$ *defined as* (3.4)*, when we have enough number of samples in each group as* $n_g \geq (c_g\omega(\mathcal{A}_g) + C_g\tau + C_g\sqrt{\frac{\log G}{\gamma}})^2$ *and enough total number of samples as* $n \geq (c_0\omega(\mathcal{A}_0) + C_0\tau + C_0\sqrt{\frac{\log G}{\gamma}})^2$ *where* $\tau > 0$ *and* $\gamma = \min_{g\in[G]}\gamma_g$*, with probability at least* $1 - 2\exp(-\gamma(\min_{g\in[G]}\omega(\mathcal{A}_g) + \tau)^2)$ *there exists an* $\epsilon \in (0, 1]$ *where:*

$$-\langle\mathbf{W}\boldsymbol{\delta}_0, \mathbf{D}\boldsymbol{\delta}_{1:G}\rangle \leq (1-\epsilon)\|\mathbf{W}\boldsymbol{\delta}_0\|_2\|\mathbf{D}\boldsymbol{\delta}_{1:G}\|_2, \quad \boldsymbol{\delta} \in \mathcal{H}$$

Results of Lemma 3.2 and Lemma 3.4 together suggest the following:

$$\|\mathbf{W}\boldsymbol{\delta}_0 + \mathbf{D}\boldsymbol{\delta}_{1:G}\|_2^2 \geq \epsilon(\|\mathbf{W}\boldsymbol{\delta}_0\|_2^2 + \|\mathbf{D}\boldsymbol{\delta}_{1:G}\|_2^2)$$

which is our desired decoupling. The following theorem uses the decoupling result of previous lemmas to establish the RE condition for the design matrix $\mathbf{X}$ (3.4).

Theorem 3.5 *Assume $\mathbf{x}_{gi}$ to be a sub-Gaussian random variable with $\mathbb{E}[\mathbf{x}_{gi}^T\mathbf{x}_{gi}] = \mathbf{I}_{p\times p}$ and $|||\mathbf{x}_{gi}|||_{\psi_2} \leq k$. Then, for all $\boldsymbol{\delta} \in \mathcal{H}$, when we have enough number of samples in each group as $n_g \geq (c_g\omega(\mathcal{A}_g) + C_g\tau + \sqrt{\frac{\log G}{\gamma}})^2$, and large enough total number of samples as $n \geq (c_0\omega(\mathcal{A}_0) + C_0\tau + \sqrt{\frac{\log G}{\gamma}})^2$ where τ and $\gamma_g > 0$, there exist a corresponding $\epsilon \in (0, 1]$ where with probability at least $1 - 4\exp(-\gamma(\min_{g\in[G]}\omega(\mathcal{A}_g) + \tau)^2)$ we have:*

$$\inf_{\boldsymbol{\delta}\in\mathcal{H}} \frac{1}{n}\|\mathbf{X}\boldsymbol{\delta}\|_2^2 \quad \geq \quad \epsilon\kappa_{\max}, \quad \kappa_{\max} = \max_{g\in[G]} \left(1 - \frac{c_g\omega(\mathcal{A}_g) + C_g\tau + \sqrt{\frac{\log G}{\gamma}}}{\sqrt{n_g}}\right).$$

Remark 3.6 *Note that the necessary number of samples to recover each private parameter, i.e., the sample complexity, is only $\sqrt{\frac{\log G}{\gamma}}$ worse than the known sample complexity of structured linear regression [14].*

Remark 3.7 *Theorem 3.5 establishes the relation between the recovery condition for the shared parameter $\boldsymbol{\beta}_0^*$ and the total number of samples n. It characterizes the exact total number of samples that are necessary to recover $\boldsymbol{\beta}_0^*$ and interestingly the sample complexity is only $\sqrt{\frac{\log G}{\gamma}}$ worse than the case of structured linear regression with single parameter, i.e., $\mathbf{y} = \mathbf{W}\boldsymbol{\beta}_0^* + \boldsymbol{\omega}$ [14].*

3.4 General Error Bound

In this section, we provide a high probability upper bound for the estimation error of the shared and private components under general norm $R(\cdot)$. Theorem 3.9 establishes a high probability upper bound for the deterministic bound of Theorem 3.1, i.e., $\frac{1}{n}2\boldsymbol{\omega}^T\mathbf{X}\mathbf{u}$, in terms of the Gaussian width of the spherical caps corresponding to each error cone, i.e., $\omega(\mathcal{C}_g \cap \mathbb{S}^{p-1})$. Following lemma provides us with the results necessary to prove Theorem 3.9.

Lemma 3.8 *For $\mathbf{x}_{gi}$ defined in Theorem 3.5 and $\boldsymbol{\omega}$ consists of i.i.d. centered unit-variance sub-Gaussian elements with $|||\omega_{gi}|||_{\psi_2} \leq K$, with probability at least $1 - \frac{\sigma_g}{G}\exp\left(-\min\left[\nu_g n_g - \log G, \frac{t^2}{\eta_g^2 k^2}\right]\right)$ we have:*

$$\|\boldsymbol{\omega}_g\|_2 \sup_{\mathbf{u}_g\in\mathcal{A}_g} \langle \mathbf{X}_g^T \frac{\boldsymbol{\omega}_g}{\|\boldsymbol{\omega}_g\|_2}, \mathbf{u}_g\rangle > \sqrt{(2K^2+1)n_g}\left(\zeta_g k\omega(\mathcal{A}_g) + \rho_g\sqrt{\log G} + \tau\right)$$

Theorem 3.9 *Assume $\mathbf{x}_{gi}$ to be a sub-Gaussian random variable with $\mathbb{E}[\mathbf{x}_{gi}^T\mathbf{x}_{gi}] = \mathbf{I}_{p\times p}$ and $|||\mathbf{x}_{gi}|||_{\psi_2} \leq k$ and $\boldsymbol{\omega}$ consists of i.i.d. centered unit-variance sub-Gaussian elements with $|||\omega_{gi}|||_{\psi_2} \leq K$, with probability at least $1 - \sigma \exp\left(-\min_{g\in[G]}\left[\nu_g n_g - \log G, \frac{\tau^2}{\eta_g^2 k^2}\right]\right)$ we have:*

$$\frac{2}{n}\boldsymbol{\omega}^T\mathbf{X}\boldsymbol{\delta} \;\leq\; \sqrt{\frac{8K^2+4}{n}}\max_{g\in[G]}\left(\zeta_g k\omega(\mathcal{A}_g) + \rho_g\sqrt{\log G} + \tau\right)$$

The following corollary characterizes the general error bound and results from the direct combination of Theorem 3.1, Theorem 3.5, and Theorem 3.9.

Corollary 3.10 *For isotropic sub-Gaussian $\mathbf{x}_{gi}$ with $|||\mathbf{x}_{gi}|||_{\psi_2} \leq k$ and i.i.d. centered unit-variance sub-Gaussian noise with $|||\omega_{gi}|||_{\psi_2} \leq K$ when we have $\forall g \in [G] : n_g \geq (c_g\omega(\mathcal{A}_g) + C_g\tau + \sqrt{\frac{\log G}{\gamma}})^2$ which lead to $\kappa = \epsilon\kappa_{\max} > 0$, the following general error bound holds with probability at least$1 - \sigma_g \exp\left(-\min_{g\in[G]}\left[\nu_g n_g - \log G, \frac{\tau^2}{\eta_g^2 k^2}\right]\right)$ for estimator* (3.2)*:*

$$\sum_{g=0}^{G}\sqrt{\frac{n_g}{n}}\|\boldsymbol{\delta}_g\|_2 \;\leq\; \frac{kC\max_{g\in[G]}\omega(\mathcal{C}_g\cap\mathbb{S}^{p-1}) + \rho\sqrt{\log G} + \tau}{\sqrt{n}} \tag{3.8}$$

where $C = (8K^2+4)\max_{g\in[G]}\zeta_g$ and $\rho = \max_{g\in[G]}\rho_g$.

Corollary 3.11 *Note that from* (3.8) *one can immediately entail the error bound for estimation of the shared parameter and all private ones as follows:*

$$\forall g \in [G] : \|\boldsymbol{\delta}_g\|_2 \leq \frac{kC\max_{g\in[G]}\omega(\mathcal{C}_g\cap\mathbb{S}^{p-1}) + \rho\sqrt{\log G} + \tau}{\sqrt{n_g}}$$

Remark 3.12 *Comparing the result of Corollary 3.11 with the case of regression with the single structured parameter $\boldsymbol{\beta}_g^*$ is interesting. Based on Corollary 3.11, $\|\boldsymbol{\delta}_g\|_2 = O((\max_{g\in[G]}\omega(\mathcal{A}_g) + \sqrt{\log G})/\sqrt{n_g})$ while sharp error bound for the single regression with $\boldsymbol{\beta}_g^*$ is $\|\boldsymbol{\delta}_g\|_2 = O(\omega(\mathcal{A}_g)/\sqrt{n_g})$. So basically by solving a more complicated data sharing model we only pay a price of $\big(\max_{g\in[G]}\omega(\mathcal{A}_g) - \omega(\mathcal{A}_g) + \sqrt{\log G}\big)/\sqrt{n_g}$ in estimation error, and $O(\log G)$ in sample complexity.*

Remark 3.13 *On the other hand, without any direct observation regarding the parameter $\boldsymbol{\beta}_0^*$ we exploit all of the groups data and get the decay rate of $1/\sqrt{n}$ for $\|\boldsymbol{\delta}_0\|_2$ by only paying a price of $\big(\max_{g\in[G]}\omega(\mathcal{A}_g) - \omega(\mathcal{A}_0) + \sqrt{\log G}\big)/\sqrt{n}$ in estimation error, and $O(\log G)$ in sample complexity of the total number of samples n.*

Remark 3.14 *For the case of sparsity, assume that each $\boldsymbol{\beta}_g^*$ is s_g-sparse and $s_0 \geq s_g$, i.e., the shared parameter is the densest. Then we have the following error bounds with high probability:*

$$\|\boldsymbol{\delta}_0\|_2 \leq c\sqrt{\frac{s_0 \log p + \sqrt{\log G}}{n}}, \quad \|\boldsymbol{\delta}_g\|_2 \leq c\sqrt{\frac{s_0 \log p + \sqrt{\log G}}{n_g}}$$

Note that here the recovery of the shared parameter is at most $c\sqrt{\frac{\sqrt{\log G}}{n}}$ worse than the case of single regression with $\boldsymbol{\beta}_0$ as the parameter. Also for the private parameters, the bound is only $c\frac{(\sqrt{s_0}-\sqrt{s_g})\sqrt{\log p}+\sqrt{\sqrt{\log G}}}{\sqrt{n_g}}$ weaker than the case of single regression.

3.5 Estimation Algorithm

We propose a projected gradient descent-like algorithm, where the corresponding private step-sizes appear as scalings in the update of the shared parameter. Therefore, we call our proposed algorithm Scaled Projected Gradient Descent (SPGD).

Algorithm 1 SPGD: Scaled Projected Gradient Descent

```
1: input: $\mathbf{X}, \mathbf{y}, (\mu_0, \ldots, \mu_G), \boldsymbol{\beta}^{(1)} = \mathbf{0}$
2: output: $\hat{\boldsymbol{\beta}}$
3: for t = 1 to T do
4:   for g=1 to G do
5:     $\boldsymbol{\beta}_g^{(t+1)} = \Pi_{\Omega_{R_g}}\left(\boldsymbol{\beta}_g^{(t)} + \mu_g \mathbf{X}_g^T\left(\mathbf{y}_g - \mathbf{X}_g\left(\boldsymbol{\beta}_0^{(t)} + \boldsymbol{\beta}_g^{(t)}\right)\right)\right)$
6:   end for
7:   $\boldsymbol{\beta}_0^{(t+1)} = \Pi_{\Omega_{R_0}}\left(\boldsymbol{\beta}_0^{(t)} + \mu_0 \mathbf{W}^T\left(\mathbf{y} - \mathbf{W}\boldsymbol{\beta}_0^{(t)} - \begin{pmatrix}\mu_1 \mathbf{X}_1 \boldsymbol{\beta}_1^{(t)} \\ \vdots \\ \mu_G \mathbf{X}_G \boldsymbol{\beta}_G^{(t)}\end{pmatrix}\right)\right)$
8: end for
```

Note that the SPGD algorithm is not exactly PGD, because of μ_gs in the second update of line 7. Also $\boldsymbol{\beta}_0^{(t+1)}$ update of line 7, is using $\boldsymbol{\beta}_g^{(t)}$ instead of the most recent value of other parameters, $\boldsymbol{\beta}_g^{(t+1)}$, hence, our algorithm is not block coordinate descent either. The closest method to SPGD is the one presented in [74], where authors show linear convergence rate for their proposed projected gradient descent method for the constraint OLS objective. In the following we show that the proposed SPGD algorithm has linear convergence rate.

3.5.1 Convergence Rate Analysis

In this section we want to upper bound the error of each iteration of the SPGD algorithm. Let's $\boldsymbol{\delta}^t = \boldsymbol{\beta}^t - \boldsymbol{\beta}^*$ be the error of iteration t of SPGD, i.e., the distance from the true parameter (not the optimization minimum, $\hat{\boldsymbol{\beta}}$). The goal of this section is to show that $\|\boldsymbol{\delta}^t\|_2$ decreases exponentially fast in t to the statistical error $\|\boldsymbol{\delta}\|_2 = \|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2$. In other word, we show that the optimization error $\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^t\|_2$ linearly converges to zero. We first start with the required definitions for our analysis.

Definition 3.15 *We define the following constants, where for simplification we assume* $\mathbf{X}_0 = \mathbf{W}$ *and* $\mathbf{w}_0 = \mathbf{w}$*:*

$$\begin{aligned}
\rho_g(\mu_g) &= \sup_{\mathbf{u},\mathbf{v} \in \mathcal{B}_g} \mathbf{v}^T \big(\mathbf{I}_g - \mu_g \mathbf{X}_g^T \mathbf{X}_g\big)\mathbf{u}, \quad g \in [G] \\
\xi_g(\mu_g) &= \mu_g \sup_{\mathbf{v} \in \mathcal{B}_g} \mathbf{v}^T \mathbf{X}_g^T \frac{\mathbf{w}_g}{\|\mathbf{w}_g\|_2}, \quad g \in [G] \\
\eta_g(\mu_g) &= \mu_g \sup_{\mathbf{v} \in \mathcal{B}_g, \mathbf{u} \in \mathcal{B}_0} -\mathbf{v}^T \mathbf{X}_g^T \mathbf{X}_g \mathbf{u}, \quad g \in [G]_{\setminus}
\end{aligned}$$

where $\mathcal{B}_g = \mathcal{C}_g \cap \mathbb{B}^p$ *is the intersection of the error cone and the unit ball.*

In the following lemma we establish a recursive relation between errors of consecutive iterations which leads to a bound for the tth iteration.

Lemma 3.16 *We have the following recursive dependency between the error of* $t+1$*th iteration and* t*th iteration:*

$$\begin{aligned}
\|\boldsymbol{\delta}_g^{(t+1)}\|_2 &\leq \rho_g(\mu_g)\|\boldsymbol{\delta}_g^t\|_2 + \xi_g(\mu_g)\|\mathbf{w}_g\|_2 + \eta_g(\mu_g)\|\boldsymbol{\delta}_0^t\|_2 \\
\|\boldsymbol{\delta}_0^{(t+1)}\|_2 &\leq \rho_0(\mu_0)\|\boldsymbol{\delta}_0^t\|_2 + \xi_0(\mu_0)\|\mathbf{w}\|_2 + \mu_0 \sum_{g=1}^{G} \eta_g(\mu_g)\|\boldsymbol{\delta}_g^t\|_2
\end{aligned}$$

From Lemma 3.16 we have:

$$\sum_{g=0}^{G} \|\boldsymbol{\delta}_g^{t+1}\|_2 \leq \left(\rho_0 + \sum_{g=1}^{G} \eta_g\right) \|\boldsymbol{\delta}_0^t\|_2 + \sum_{g=1}^{G} \left(\rho_g + \mu_0 \eta_g\right) \|\boldsymbol{\delta}_g^t\|_2 + \sum_{g=0}^{G} \xi_g \|\mathbf{w}_g\|_2 \tag{3.9}$$

By recursively applying the inequality (3.9), we can easily derive the following theorem for the upper bound of error in each iteration.

Theorem 3.17 *For $\mathbf{x} \in \mathbb{R}^p$, any norm $R(\cdot)$, and initialization $\boldsymbol{\beta}^{(0)}$ of the SPGD 3. We have the following bound for error at iteration $t+1$ of SPGD:*

$$\|\boldsymbol{\delta}^{t+1}\|_2 \quad \leq \quad \alpha^t \sum_{g=0}^{G} \|\boldsymbol{\beta}_g^*\|_2 + \frac{1-\alpha^t}{1-\alpha} \sum_{g=0}^{G} \xi_g(\mu_g)\|\mathbf{w}_g\|_2, \tag{3.10}$$

where $\alpha = \max\Big(\max_{g \in [G]_\setminus}(\rho_g + \mu_0 \eta_g), \rho_0 + \sum_{g=1}^G \eta_g\Big)$.

The RHS of (3.10) consists of two term. If we keep $\alpha < 1$, the first term approaches zero exponentially fast, i.e., with linear rate. The first term corresponds to the optimization error, i.e., $\|\boldsymbol{\beta}^{t+1} - \hat{\boldsymbol{\beta}}\|_2$. We will show that the second term approximates the upper bound for statistical error, i.e., $\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2 = \|\boldsymbol{\delta}\|_2 = \sqrt{\sum_{g=1}^G \|\boldsymbol{\delta}_g\|_2^2}$ where we characterized a scaled version of it as $\sum_{g=1}^G \sqrt{\frac{n_g}{n}} \|\boldsymbol{\delta}_g\|_2$ in (3.8). Therefore, if we can make $\alpha < 1$, the estimation error of SPGD algorithm linearly converges to the approximate statistical error bound.

One way for having $\alpha < 1$ is to keep the coefficients of all $\|\boldsymbol{\delta}_g^t\|_2$ in (3.9) strictly below one, simultaneously:

$$\left(\rho_0(\mu_0) + \sum_{g=1}^{G} \eta_g(\mu_g)\right) \quad < \quad 1 \tag{3.11}$$

$$\forall g \in [G]_\setminus : (\rho_g(\mu_g) + \mu_0 \eta_g(\mu_g)) \quad < \quad 1 \tag{3.12}$$

To this end, we first establish high probability upper bound for ρ_g and η_g and then try to keep the coefficients of $\|\boldsymbol{\delta}_g^t\|_2$s below one.

Lemma 3.18 *We can establish the following high probability upper bounds for the coefficients:*

$$\begin{aligned} \rho_g(\mu_g) \leq & 1 - \mu_g d_g, \quad \textit{w.p.} \quad && 1 - 2\exp\left(-\gamma_g(\omega(\mathcal{A}_g) + \tau)^2\right) \\ \eta_g(\mu_g) \leq & \quad\ \mu_g s_g, \quad \textit{w.p.} \quad && 1 - 4\exp\Big(-\gamma\left(\min(\omega(\mathcal{A}_0), \omega(\mathcal{A}_g)) + \tau\right)^2\Big) \end{aligned}$$

where $d_g = n_g - c_g\sqrt{n_g}\omega(\mathcal{A}_g) - C_g\sqrt{n_g}\tau$, $s_g = n_g + c_g\sqrt{n_g}\max(\omega(\mathcal{A}_0), \omega(\mathcal{A}_g)) + C_g\sqrt{n_g}\tau$, and $\gamma = \min_{g \in [G]} \gamma_g$.

The following theorem shows that for a specific set of step-sizes both (3.11) and (3.12) holds with high probability.

Theorem 3.19 *For $\tau > 0$, when per group and total number of samples are large enough, i.e., $n_g \geq \left(c_g\omega(\mathcal{A}_g) + C_g\sqrt{\frac{\log G}{\gamma}} + C_g\tau\right)^2$, and we select the following step sizes:*

$$\begin{aligned} \mu_0 &< \max_{g\in[G]_\setminus} \frac{\left(\sqrt{n_g} - c_g\omega(\mathcal{A}_g) - C_g\sqrt{\frac{\log G}{\gamma}} - C_g\tau\right)}{\left(\sqrt{n_g} + c_g\max(\omega(\mathcal{A}_0), \omega(\mathcal{A}_g)) + C_g\sqrt{\frac{\log G}{\gamma}} + C_g\tau\right)} \\ \forall g \in [G]_\setminus : \mu_g &< \frac{\mu_0\left(\sqrt{n} - c_0\omega(\mathcal{A}_0) - C_0\sqrt{\frac{\log G}{\gamma}} - C_0\tau\right)}{G\left(\sqrt{n_g} + c_g\max(\omega(\mathcal{A}_0), \omega(\mathcal{A}_g)) + C_g\sqrt{\frac{\log G}{\gamma}} + C_g\tau\right)}, \end{aligned}$$

with probability at least $1 - 10\exp\left(-\gamma\left(\min_{g\in[G]}\omega(\mathcal{A}_g) + \tau\right)^2\right)$, α of Theorem 3.17 becomes less than one and the Algorithm 3 linearly converges to the scaled error of $\frac{\sqrt{n}}{1-\alpha}\left(\max_{g\in[G]}\omega(\mathcal{A}_g) + \sqrt{\log G}\right)$

Remark 3.20 *One can readily perform the same analysis for the scaled version of the error presented in Theorem 3.1 as $\sum_{g=1}^{G}\alpha_g\|\boldsymbol{\delta}_g^t\|_2$ where $\alpha_g = \sqrt{\frac{n_g}{n}}$. The only difference is that the step-size for the shared parameter scales as $\mu_0\sqrt{\frac{n_g}{n}}$ where μ_0 is the step size of Theorem 3.19. Then the upper bound of error $\sum_{g=1}^{G}\alpha_g\|\boldsymbol{\delta}_g^t\|_2$ converges to the upper bound of the statistical error $\sum_{g=1}^{G}\alpha_g\|\boldsymbol{\delta}_g\|_2$ computed in Theorem 3.10, i.e., $c\left(\max_{g\in[G]}\omega(\mathcal{A}_g) + \sqrt{\log G}\right)/\sqrt{n}$, similar to* (3.8).

We can simplify the result of Theorem 3.19 to have a better guideline for choosing the step size. The following corollary characterizes such step sizes.

Corollary 3.21 *If the number of samples satisfies $n_g \geq \left(c_g\omega(\mathcal{A}_g) + C_g\sqrt{\frac{\log G}{\gamma}} + 1\right)^2$ then for the following step sizes SPGD algorithm linearly converges to an approximate error bound with probability at least $1 - 10\exp\left(-\gamma\left(\min_{g\in[G]}\omega(\mathcal{A}_g)\right)^2\right)$.*

$$\begin{aligned} \mu_0 &\leq \left(\sqrt{n} + c\max_{g\in[G]}\omega(\mathcal{A}_g) + C\sqrt{\log G} + 1\right)^{-1} \\ \forall g \in [G]_\setminus : \mu_g &\leq \frac{\mu_0}{G}\left(\sqrt{n_g} + c\max_{g\in[G]}\omega(\mathcal{A}_g) + C\sqrt{\log G} + 1\right)^{-1} \end{aligned}$$

Remark 3.22 *For the example of sparse shared and private parameters of Remark 3.14, where $\boldsymbol{\beta}_g^*$ is s_g-sparse and $s_0 \geq s_g$, we can pick the following step sizes to have linear convergence* *with high probability for the SPGD algorithm:*

$$\begin{aligned} \mu_0 &= \left(\sqrt{n} + c\sqrt{s_0 \log p} + C\sqrt{\log G} + 1\right)^{-1} \\ \forall g \in [G]_\setminus : \mu_g &\leq \frac{\mu_0^2}{G} \end{aligned}$$

3.6 Experiment

In this section we supplement our theoretical results with a simple synthetic experiment. We focus on the case of two groups, i.e., $G = 2$. The dimension $p = 1000$ and the structure is sparsity induced by l_1-norm. The parameters $\boldsymbol{\beta}_0^*$, $\boldsymbol{\beta}_1^*$, and $\boldsymbol{\beta}_2^*$ are 20, 10, and 5-sparse respectively. The sparsity pattern is as follows:$\boldsymbol{\beta}_0^* = (\underbrace{1, \dots, 1}_{1-20}, 0, \dots)$,$\boldsymbol{\beta}_1^* = (\dots, 0, \underbrace{2, \dots, 2}_{51-60}, 0, \dots)$, and $\boldsymbol{\beta}_2^* = (\dots, 0, \underbrace{-2, \dots, -2}_{96-100}, 0, \dots)$.

For the distribution of input and noise we have $\mathbf{x}_{gi} \sim N(0, \sigma_x^2 \mathbf{I})$ and $\omega_{gi} \sim N(0, \sigma_w^2)$ with $\sigma_x^2 = .3$ and $\sigma_w^2 = .1$. We use the SPGD method (Algorithm 1) to solve the optimization problem (3.2). The projection to the l_1 ball can be efficiently performed by the method proposed in [75].

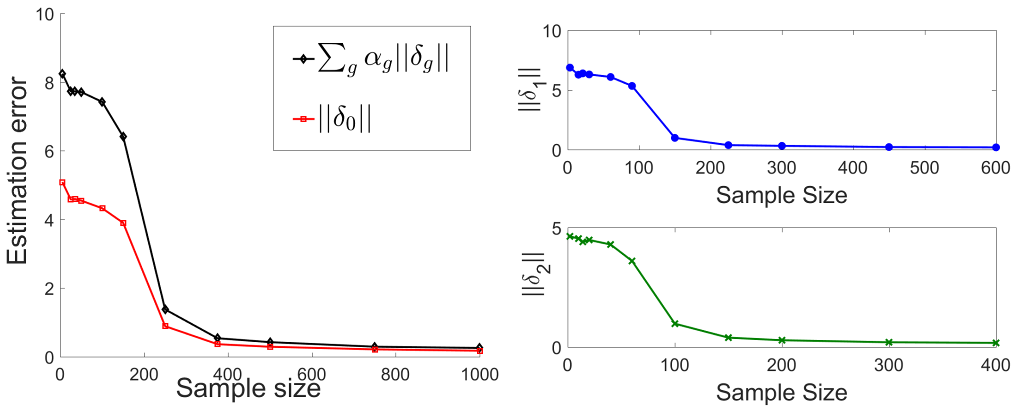
While changing n in the experiments, we keep the ratio $\frac{n_1}{n_2} = \frac{2}{3}$ fixed. Figure 3.1b shows the per-group error for different sample size which follows $1/\sqrt{n_g}$ decay. Finally, Figure 3.1a shows the decay of the error as sample size increases for the shared component recovery and the error for summation of the form (1.14). As expected errors decay as $1/\sqrt{n}$.

3.7 Proofs

3.7.1 Proof of Theorem 3.1

Starting from (3.5), for the lower bound we get:

$$\begin{aligned} \frac{1}{n}\|\mathbf{X}\boldsymbol{\delta}\|_2^2 &\geq \frac{1}{n}\inf_{\mathbf{u}\in\mathcal{H}} \|\mathbf{X}\mathbf{u}\|_2^2 \left(\sum_{g=0}^{G} \alpha_g \|\boldsymbol{\delta}_g\|_2\right)^2 \qquad (3.13) \\ &\geq \kappa \left(\sum_{g=0}^{G} \alpha_g \|\boldsymbol{\delta}_g\|_2\right)^2 \end{aligned}$$



(a) Estimation error for the shared parameter $\boldsymbol{\beta}_0^*$.

(b) Estimation error for private group parameters $\boldsymbol{\beta}_1^*$ and $\boldsymbol{\beta}_2^*$.

Figure 3.1: Estimation error with different sample size. 3.1a compares the error with the LHS of (1.14). Each point on the diagram is an average over 10 experiments.

where $0 < \kappa \leq \frac{1}{n} \inf_{\mathbf{u} \in \mathcal{H}} \|\mathbf{X}\mathbf{u}\|_2^2$ is known as Restricted Eigenvalue (RE) condition. The upper bound will factorize as follows:

$$\frac{2}{n}\boldsymbol{\omega}^T\mathbf{X}\boldsymbol{\delta} \quad \leq \quad \frac{2}{n}\boldsymbol{\omega}^T\mathbf{X}\mathbf{u}\left(\sum_{g=0}^{G}\alpha_g\|\boldsymbol{\delta}_g\|_2\right), \quad \mathbf{u} \in \mathcal{H} \tag{3.14}$$

Putting together both inequalities (3.13) and (3.14) completes the proof.

3.7.2 Proof of Lemma 3.4

In the following we show that (3.8) holds with high probability as long as we have enough number of samples both in each group and totally. The LHS of (3.8) is equal to $-\sum_{g=1}^{G}\langle \mathbf{X}_g\boldsymbol{\delta}_0, \mathbf{X}_g\boldsymbol{\delta}_g\rangle$, to which we can apply the Cauchy-Shwarz inequality and get:

$$-\langle \mathbf{W}\boldsymbol{\delta}_0, \mathbf{D}\boldsymbol{\delta}_{1:G}\rangle = -\sum_{g=1}^{G}\langle \mathbf{X}_g\boldsymbol{\delta}_0, \mathbf{X}_g\boldsymbol{\delta}_g\rangle \leq \sum_{g=1}^{G}\|\mathbf{X}_g\boldsymbol{\delta}_0\|_2\|\mathbf{X}_g\boldsymbol{\delta}_g\|_2$$

So the problem reduces to finding the $0 \leq 1 - \epsilon < 1$ satisfying:

$$\begin{aligned} \sum_{g=1}^{G} \|\mathbf{X}_g \boldsymbol{\delta}_0\|_2 \|\mathbf{X}_g \boldsymbol{\delta}_g\|_2 &\leq (1-\epsilon)\|\mathbf{W}\boldsymbol{\delta}_0\|_2 \|\mathbf{D}\boldsymbol{\delta}_{1:G}\|_2 && (3.15) \\ &= (1-\epsilon)\sqrt{\left(\sum_{g=1}^{G} \|\mathbf{X}_g \boldsymbol{\delta}_0\|_2^2\right)\left(\sum_{g=1}^{G} \|\mathbf{X}_g \boldsymbol{\delta}_g\|_2^2\right)} && (3.16) \\ &= (1-\epsilon)\sqrt{\sum_{i,j=1}^{G} \left(\|\mathbf{X}_i \boldsymbol{\delta}_0\|_2 \|\mathbf{X}_j \boldsymbol{\delta}_j\|_2\right)^2} \end{aligned}$$

A simple upper bound for the RHS can be derived from $\sqrt{\sum_i a_i^2} \leq \sum_i a_i$, for $a_i \geq 0$. To have $1 - \epsilon$ strictly less than one, we need and strict inequality. More specifically if $0 < \sqrt{\sum_i a_i^2}$ then at least one of the a_i is non-zero, which leads to a strict inequality. In other words, $\forall a_i \geq 0, \sqrt{\sum_i a_i^2} > 0 : \sqrt{\sum_i a_i^2} < \sum_i a_i$. So let's assume $\|\mathbf{W}\boldsymbol{\delta}_0\|_2 \|\mathbf{D}\boldsymbol{\delta}_{1:G}\|_2 > 0$ and use the strict inequality $\sqrt{\sum_i a_i^2} < \sum_i a_i$:

$$\begin{aligned} \sum_{g=1}^{G} \|\mathbf{X}_g \boldsymbol{\delta}_0\|_2 \|\mathbf{X}_g \boldsymbol{\delta}_g\|_2 &\leq (1-\epsilon)\sqrt{\sum_{i,j=1}^{G} \left(\|\mathbf{X}_i \boldsymbol{\delta}_0\|_2 \|\mathbf{X}_j \boldsymbol{\delta}_j\|_2\right)^2} \\ &< (1-\epsilon)\sum_{i,j=1}^{G} \|\mathbf{X}_i \boldsymbol{\delta}_0\|_2 \|\mathbf{X}_j \boldsymbol{\delta}_j\|_2 \end{aligned}$$

The terms of LHS are subset of terms on RHS summation, so if $\|\mathbf{W}\boldsymbol{\delta}_0\|_2 \|\mathbf{D}\boldsymbol{\delta}_{1:G}\|_2$ is bounded away from zero, there always exist an $0 < \epsilon \leq 1$ for which we have:

$$\frac{\sum_{g=1}^{G} \|\mathbf{X}_g \boldsymbol{\delta}_0\|_2 \|\mathbf{X}_g \boldsymbol{\delta}_g\|_2}{\sum_{i,j=1}^{G} \|\mathbf{X}_i \boldsymbol{\delta}_0\|_2 \|\mathbf{X}_j \boldsymbol{\delta}_j\|_2} \leq (1-\epsilon) < 1 \tag{3.17}$$

Now we show that the assumption $\|\mathbf{W}\boldsymbol{\delta}_0\|_2 \|\mathbf{D}\boldsymbol{\delta}_{1:G}\|_2 > 0$ holds with high probability for enough number of samples. It is equivalent to show that the square of terms are bounded away from zero.

$$\begin{aligned} \|\mathbf{W}\boldsymbol{\delta}_0\|_2^2 \|\mathbf{D}\boldsymbol{\delta}_{1:G}\|_2^2 &\geq \|\mathbf{W}\boldsymbol{\delta}_0\|_2^2 \left(\sum_{g=1}^{G} \|\mathbf{X}_g \boldsymbol{\delta}_g\|_2^2\right), \quad \boldsymbol{\delta} \in \mathcal{A} \\ &\geq \|\boldsymbol{\delta}_0\|_2 \inf_{\mathbf{u}_0 \in \mathcal{C}_0 \cap \mathbb{S}^{p-1}} \|\mathbf{W}\mathbf{u}_0\|_2^2 \left(\sum_{g=1}^{G} \|\boldsymbol{\delta}_g\|_2 \inf_{\mathbf{u}_g \in \mathcal{C}_g \cap \mathbb{S}^{p-1}} \|\mathbf{X}_g \mathbf{u}_g\|_2^2\right) && (3.18) \end{aligned}$$

To avoid cluttering, we name $\mathcal{A}_g = \mathcal{C}_g \cap \mathbb{S}^{p-1}$ the corresponding spherical cap of the error cone. We want to bound the RHS of (3.18) away from zero, so we call the following event, the bad event:

$$\mathcal{E} = \inf_{\mathbf{u}_0 \in \mathcal{A}_0} \|\mathbf{W}\mathbf{u}_0\|_2^2 \left(\sum_{g=1}^{G} \inf_{\mathbf{u}_g \in \mathcal{A}_g} \|\mathbf{X}_g \mathbf{u}_g\|_2^2 \right) < n\kappa_0(\tau) \sum_{g=1}^{G} n_g \kappa_g(\tau). \tag{3.19}$$

where $\kappa_g(\tau) = 1 - c_g \frac{\omega(\mathcal{A}_g)}{\sqrt{n_g}} - C_g \frac{\tau}{\sqrt{n_g}}$ and for enough number of samples $\kappa_g > 0$. Remember the following result from Lemma 3.3:

$$\inf_{\boldsymbol{\delta}_g \in \mathcal{A}_g} \|\mathbf{X}_g \boldsymbol{\delta}_g\|_2^2 \quad \geq \quad n_g \kappa_g(\tau), \quad \text{w.p. } (1 - 2\exp(-\gamma_g(\omega(\mathcal{A}_g) + \tau)^2) \tag{3.20}$$

In the following with the help of Lemma 3.3 we upper bound the probability of the bad event happening using the law of total probability.

$$\begin{aligned}
& & & \mathbb{P}\left(\inf_{\mathbf{u}_0 \in \mathcal{A}_0} \|\mathbf{W}\mathbf{u}_0\|_2^2 \left(\sum_{g=1}^{G} \inf_{\mathbf{u}_g \in \mathcal{A}_g} \|\mathbf{X}_g \mathbf{u}_g\|_2^2 \right) < n\kappa_0(\tau) \sum_{g=1}^{G} n_g \kappa_g(\tau) \right) \\
& & \leq & \mathbb{P}\left(\inf_{\mathbf{u}_0 \in \mathcal{A}_0} \|\mathbf{W}\mathbf{u}_0\|_2^2 < n\kappa_0(\tau) \right) + \mathbb{P}\left(\sum_{g=1}^{G} \inf_{\mathbf{u}_g \in \mathcal{A}_g} \|\mathbf{X}_g \mathbf{u}_g\|_2^2 < \sum_{g=1}^{G} n_g \kappa_g(\tau) \right) \\
& & \leq & \mathbb{P}\left(\inf_{\mathbf{u}_0 \in \mathcal{A}_0} \|\mathbf{W}\mathbf{u}_0\|_2^2 < n\kappa_0(\tau) \right) + \mathbb{P}\left(\sum_{g=2}^{G} \inf_{\mathbf{u}_g \in \mathcal{A}_g} \|\mathbf{X}_g \mathbf{u}_g\|_2^2 < \sum_{g=2}^{G} n_g \kappa_g(\tau) \right) \\
& & + & \mathbb{P}\left(\inf_{\mathbf{u}_2 \in \mathcal{A}_2} \|\mathbf{X}_2 \mathbf{u}_2\|_2^2 < n_2 \kappa_2(\tau) \right) \\
& (\mathbf{X}_0 = \mathbf{W}, n_0 = n) & \leq & \sum_{g=0}^{G} \mathbb{P}\left(\inf_{\mathbf{u}_g \in \mathcal{A}_g} \|\mathbf{X}_g \mathbf{u}_g\|_2^2 < n_g \kappa_g(\tau) \right) \\
& (3.20) & \leq & G \max_{g \in [G]} \mathbb{P}\left(\inf_{\mathbf{u}_g \in \mathcal{A}_g} \|\mathbf{X}_g \mathbf{u}_g\|_2^2 < n_g \kappa_g(\tau) \right) \\
& & \leq & 2G \exp(-\gamma(\min_{g \in [G]} \omega(\mathcal{A}_g) + \tau)^2)
\end{aligned}$$

where $\gamma = \min_{g \in [G]} \gamma_g$. Now to remove the multiplicative G in the probability we set $\tau = a + \sqrt{\frac{\log G}{\gamma}}$. This translates to requiring slightly more number samples, because to keep $\kappa_g > 0$ we need $n_g \geq (c_g \omega(\mathcal{A}_g) + C_g a + \sqrt{\frac{\log G}{\gamma}})^2$. ■

3.7.3 Proof of Theorem 3.5

As shown in Lemma 3.4 the assumption of Lemma 3.2 holds with high probability. Therefore we have the conclusion of the Lemma 3.2 with high probability. More concretely, for the bad event $\mathcal{E}$ of (3.19), when $\neg\mathcal{E}$ happens we have:

$$\|\mathbf{W}\boldsymbol{\delta}_0 + \mathbf{D}\boldsymbol{\delta}_{1:G}\|_2^2 \geq \epsilon \left(\|\mathbf{W}\boldsymbol{\delta}_0\|_2^2 + \|\mathbf{D}\boldsymbol{\delta}_{1:G}\|_2^2\right)$$

So we can write the following bound:

$$
\begin{aligned}
& \mathbb{P}\left(\frac{1}{n}\inf_{\boldsymbol{\delta}\in\mathcal{H}}\|\mathbf{X}\boldsymbol{\delta}\|_2^2 \leq \epsilon\sum_{g=0}^{G}\frac{n_g}{n}\|\boldsymbol{\delta}_g\|_2^2\kappa_g\right) \\
= \;& \mathbb{P}\left(\|\mathbf{W}\boldsymbol{\delta}_0+\mathbf{D}\boldsymbol{\delta}_{1:G}\|_2^2 \leq \epsilon\sum_{g=0}^{G}n_g\|\boldsymbol{\delta}_g\|_2^2\kappa_g\right) \\
= \;& \mathbb{P}\left(\|\mathbf{W}\boldsymbol{\delta}_0+\mathbf{D}\boldsymbol{\delta}_{1:G}\|_2^2 \leq \epsilon\sum_{g=0}^{G}n_g\|\boldsymbol{\delta}_g\|_2^2\kappa_g\middle|\mathcal{E}\right)\mathbb{P}(\mathcal{E}) \\
+ \;& \mathbb{P}\left(\|\mathbf{W}\boldsymbol{\delta}_0+\mathbf{D}\boldsymbol{\delta}_{1:G}\|_2^2 \leq \epsilon\sum_{g=0}^{G}n_g\|\boldsymbol{\delta}_g\|_2^2\kappa_g\middle|\neg\mathcal{E}\right)\mathbb{P}(\neg\mathcal{E}) \\
\leq \;& \mathbb{P}\left(\epsilon\left(\|\mathbf{W}\boldsymbol{\delta}_0\|_2^2+\|\mathbf{D}\boldsymbol{\delta}_{1:G}\|_2^2\right) \leq \epsilon\sum_{g=0}^{G}n_g\|\boldsymbol{\delta}_g\|_2^2\kappa_g\right)+\mathbb{P}(\mathcal{E}) \\
= \;& \mathbb{P}\left(\left(\|\mathbf{W}\boldsymbol{\delta}_0\|_2^2+\sum_{g=1}^{G}\|\mathbf{X}_g\boldsymbol{\delta}_g\|_2^2\right) \leq \sum_{g=0}^{G}n_g\|\boldsymbol{\delta}_g\|_2^2\kappa_g\right)+\mathbb{P}(\mathcal{E}) \\
\leq \;& \mathbb{P}\left(\left(\|\boldsymbol{\delta}_0\|_2^2\inf_{\mathbf{u}\in\mathcal{A}_0}\|\mathbf{W}\mathbf{u}_0\|_2^2+\|\boldsymbol{\delta}_g\|_2^2\sum_{g=1}^{G}\inf_{\mathbf{u}\in\mathcal{A}_g}\|\mathbf{X}_g\mathbf{u}_g\|_2^2\right) \leq \sum_{g=0}^{G}n_g\|\boldsymbol{\delta}_g\|_2^2\kappa_g\right)+\mathbb{P}(\mathcal{E}) \\
\leq \;& \mathbb{P}\left(\sum_{g=0}^{G}\|\boldsymbol{\delta}_g\|_2^2\inf_{\mathbf{u}\in\mathcal{A}_g}\|\mathbf{X}_g\mathbf{u}_g\|_2^2 \leq \sum_{g=0}^{G}n_g\|\boldsymbol{\delta}_g\|_2^2\kappa_g\right)+\mathbb{P}(\mathcal{E}) \\
\leq \;& \mathbb{P}\left(\sum_{g=0}^{G}\frac{1}{n_g}\inf_{\mathbf{u}\in\mathcal{A}_g}\|\mathbf{X}_g\mathbf{u}_g\|_2^2 \leq \sum_{g=0}^{G}\kappa_g\right)+\mathbb{P}(\mathcal{E}) \\
\leq \;& \sum_{g=0}^{G}\mathbb{P}\left(\frac{1}{n_g}\inf_{\mathbf{u}\in\mathcal{A}_g}\|\mathbf{X}_g\mathbf{u}_g\|_2^2 \leq \kappa_g(\tau)\right)+\mathbb{P}(\mathcal{E}) \\
\leq \;& 2G\exp(-\gamma(\min_{g\in[G]}\omega(\mathcal{A}_g)+\tau)^2)+2G\exp(-\gamma(\min_{g\in[G]}\omega(\mathcal{A}_g)+\tau)^2) \\
\leq \;& 4G\exp(-\gamma(\min_{g\in[G]}\omega(\mathcal{A}_g)+\tau)^2)
\end{aligned}
$$

where $\gamma = \min_{g\in[G]}\gamma_g$ and $\epsilon \in (0,1]$. Like before, to remove the multiplicative G in the probability we set $\tau = a + \sqrt{\frac{\log G}{\gamma}}$. This translates to requiring slightly more number samples, because to keep $\kappa_g > 0$ we need $n_g \geq (c_g\omega(\mathcal{A}_g) + C_g a + \sqrt{\frac{\log G}{\gamma}})^2$.

Therefore we have:

$$\begin{aligned}
1 - 4\exp(-\gamma(\min_{g\in[G]} \omega(\mathcal{A}_g) + \tau)^2) &\geq \mathbb{P}\left(\frac{1}{n}\inf_{\boldsymbol{\delta}\in\mathcal{H}} \|\mathbf{X}\boldsymbol{\delta}\|_2^2 \geq \epsilon \sum_{g=0}^{G} \frac{n_g}{n}\|\boldsymbol{\delta}_g\|_2^2 \kappa_g\right) \\
(\kappa_{\max} = \max_{g\in[G]} \kappa_g) &\geq \mathbb{P}\left(\frac{1}{n}\inf_{\boldsymbol{\delta}\in\mathcal{H}} \|\mathbf{X}\boldsymbol{\delta}\|_2^2 \geq \epsilon\kappa_{\max} \sum_{g=0}^{G} \frac{n_g}{n}\|\boldsymbol{\delta}_g\|_2^2\right) \\
(\sqrt{\frac{n_g}{n}}\|\boldsymbol{\delta}_g\|_2 \geq \frac{n_g}{n}\|\boldsymbol{\delta}_g\|_2^2) &\geq \mathbb{P}\left(\frac{1}{n}\inf_{\boldsymbol{\delta}\in\mathcal{H}} \|\mathbf{X}\boldsymbol{\delta}\|_2^2 \geq \epsilon\kappa_{\max} \sum_{g=0}^{G} \sqrt{\frac{n_g}{n}}\|\boldsymbol{\delta}_g\|_2\right) \\
(\boldsymbol{\delta} \in \mathcal{H}) &= \mathbb{P}\left(\frac{1}{n}\inf_{\boldsymbol{\delta}\in\mathcal{H}} \|\mathbf{X}\boldsymbol{\delta}\|_2^2 \geq \epsilon\kappa_{\max}\right)
\end{aligned}$$

■

3.7.4 Proof of Lemma 3.8

To avoid cluttering let $f_g(\boldsymbol{\omega}_g, \mathbf{X}_g) = \sqrt{\frac{n}{n_g}}\|\boldsymbol{\omega}_g\|_2 \sup_{\mathbf{u}_g\in\mathcal{A}_g} \langle \mathbf{X}_g^T \frac{\boldsymbol{\omega}_g}{\|\boldsymbol{\omega}_g\|_2}, \mathbf{u}_g\rangle$, $h_g = \zeta_g k\omega(\mathcal{A}_g) + \rho_g\sqrt{\log G} + \tau$, where $r_g = \sqrt{\frac{n}{n_g}}\sqrt{(2K^2+1)n_g}$.

$$\begin{aligned}
& \mathbb{P}\left(f_g(\boldsymbol{\omega}_g, \mathbf{X}_g) > h_g r_g\right) \qquad (3.21)\\
= \; & \mathbb{P}\left(f_g(\boldsymbol{\omega}_g, \mathbf{X}_g) > h_g r_g \Big| \sqrt{\frac{n}{n_g}}\|\boldsymbol{\omega}_g\|_2 > r_g\right)\mathbb{P}\left(\sqrt{\frac{n}{n_g}}\|\boldsymbol{\omega}_g\|_2 > r_g\right) \\
+ \; & \mathbb{P}\left(f_g(\boldsymbol{\omega}_g, \mathbf{X}_g) > h_g r_g \Big| \sqrt{\frac{n}{n_g}}\|\boldsymbol{\omega}_g\|_2 < r_g\right)\mathbb{P}\left(\sqrt{\frac{n}{n_g}}\|\boldsymbol{\omega}_g\|_2 < r_g\right) \\
\leq \; & \mathbb{P}\left(\sqrt{\frac{n}{n_g}}\|\boldsymbol{\omega}_g\|_2 > r_g\right) + \mathbb{P}\left(f_g(\boldsymbol{\omega}_g, \mathbf{X}_g) > h_g r_g \Big| \sqrt{\frac{n}{n_g}}\|\boldsymbol{\omega}_g\|_2 < r_g\right) \\
\leq \; & \mathbb{P}\left(\|\boldsymbol{\omega}_g\|_2 > \sqrt{(2K^2+1)n_g}\right) + \mathbb{P}\left(\sup_{\mathbf{u}_g\in\mathcal{C}_g\cap\mathbb{S}^{p-1}} \langle \mathbf{X}_g^T \frac{\boldsymbol{\omega}_g}{\|\boldsymbol{\omega}_g\|_2}, \mathbf{u}_g\rangle > h_g\right) \\
\leq \; & \mathbb{P}\left(\|\boldsymbol{\omega}_g\|_2 > \sqrt{(2K^2+1)n_g}\right) + \sup_{\mathbf{v}\in\mathbb{S}^{p-1}} \mathbb{P}\left(\sup_{\mathbf{u}_g\in\mathcal{C}_g\cap\mathbb{S}^{p-1}} \langle \mathbf{X}_g^T \mathbf{v}, \mathbf{u}_g\rangle > h_g\right)
\end{aligned}$$

We first focus on the first term. Since $\boldsymbol{\omega}_g$ consists of i.i.d. centered unit-variance sub-Gaussian elements with $|||\omega_{gi}|||_{\psi_2} < K$, ω_{gi}^2 is sub-exponential with $|||\omega_{gi}|||_{\psi_1} < 2K^2$. Let's apply the Bernstein's inequality to $\|\boldsymbol{\omega}_g\|_2^2 = \sum_{i=1}^{n_g} \omega_{gi}^2$:

$$\mathbb{P}\left(\left|\|\boldsymbol{\omega}_g\|_2^2 - \mathbb{E}\|\boldsymbol{\omega}_g\|_2^2\right| > \tau\right) \leq 2\exp\left(-\nu_g \min\left[\frac{\tau^2}{4K^4 n_g}, \frac{\tau}{2K^2}\right]\right)$$

We also know that $\mathbb{E}\|\omega_g\|_2^2 \leq n_g$ [14] which gives us:

$$\mathbb{P}\left(\|\boldsymbol{\omega}_g\|_2 > \sqrt{n_g + \tau}\right) \leq 2\exp\left(-\nu_g \min\left[\frac{\tau^2}{4K^4 n_g}, \frac{\tau}{2K^2}\right]\right) \tag{3.22}$$

Finally, we set $\tau = 2K^2 n_g$:

$$\begin{aligned} \mathbb{P}\left(\|\boldsymbol{\omega}_g\|_2 > \sqrt{(2K^2+1)n_g}\right) &\leq 2\exp\left(-\nu_g n_g\right) \qquad (3.23) \\ &= \frac{2}{G}\exp\left(-\nu_g n_g + \log G\right) \end{aligned}$$

Now we upper bound the second term of (3.21). Given any fixed $\mathbf{v} \in \mathbb{S}^{p-1}$, $\mathbf{X}_g\mathbf{v}$ is a sub-Gaussian random vector with $\left\vert\left\vert\left\vert\mathbf{X}_g^T\mathbf{v}\right\vert\right\vert\right\vert_{\psi_2} \leq C_g k$ [14]. From [14, Theorem 9] for any $\mathbf{v} \in \mathbb{S}^{p-1}$ we have:

$$\mathbb{P}\left(\sup_{\mathbf{u}_g \in \mathcal{A}_g} \langle \mathbf{X}_g^T\mathbf{v}, \mathbf{u}_g\rangle > \upsilon_g C_g k \omega(\mathcal{A}_g) + t\right) \leq \lambda_g \exp\left(-\left(\frac{t}{\theta_g C_g k \phi_g}\right)^2\right) \tag{3.24}$$

where $\phi_g = \sup_{\mathbf{u}_g \in \mathcal{A}_g} \|\mathbf{u}_g\|_2$ and in our problem $\phi_g = 1$. We now substitute $t = \tau + \rho_g\sqrt{\log G}$ where $\rho_g = \theta_g C_g k$.

$$\begin{aligned} \mathbb{P}\left(\sup_{\mathbf{u}_g \in \mathcal{A}_g} \langle \mathbf{X}_g^T\mathbf{v}, \mathbf{u}_g\rangle > \upsilon_g C_g k \omega(\mathcal{A}_g) + \rho_g\sqrt{\log G} + \tau\right) &\leq \lambda_g \exp\left(-\left(\frac{\tau + \rho_g\sqrt{\log G}}{\rho_g}\right)^2\right) \\ &\leq \lambda_g \exp\left(-\log G - \left(\frac{\tau}{\theta_g C_g k}\right)^2\right) \\ &\leq \frac{\lambda_g}{G}\exp\left(-\left(\frac{\tau}{\theta_g C_g k}\right)^2\right) \end{aligned}$$

Now let's:

$$\begin{aligned} &\mathbb{P}\left(f_g(\boldsymbol{\omega}_g, \mathbf{X}_g) > \sqrt{\frac{n}{n_g}}\sqrt{(2K^2+1)n_g}\left(\upsilon_g C_g k \omega(\mathcal{A}_g) + \rho_g\sqrt{\log G} + \tau\right)\right) \\ \leq\ &\frac{\sigma_g}{G}\exp\left(-\min\left[\nu_g n_g - \log G, \frac{t^2}{\theta_g^2 C_g^2 k^2}\right]\right) \\ \leq\ &\frac{\sigma_g}{G}\exp\left(-\min\left[\nu_g n_g - \log G, \frac{t^2}{\eta_g^2 k^2}\right]\right) \end{aligned}$$

where $\sigma_g = \lambda_g + 2$, $\zeta_g = \upsilon_g C_g$, $\eta_g = \theta_g C_g$.

3.7.5 Proof of Theorem 3.9

Before taking the expectation we massage the equation as follows:

$$\begin{aligned}
\boldsymbol{\omega}^T\mathbf{X}\boldsymbol{\delta} &= \langle \mathbf{W}^T\boldsymbol{\omega}, \boldsymbol{\delta}_0\rangle + \sum_{g=1}^{G}\langle \mathbf{X}_g^T\boldsymbol{\omega}_g, \boldsymbol{\delta}_g\rangle \\
&= \|\boldsymbol{\delta}_0\|_2\langle \mathbf{W}^T\frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|_2}, \frac{\boldsymbol{\delta}_0}{\|\boldsymbol{\delta}_0\|_2}\rangle\|\boldsymbol{\omega}\|_2 + \sum_{g=1}^{G}\|\boldsymbol{\delta}_g\|_2\langle \mathbf{X}_g^T\frac{\boldsymbol{\omega}_g}{\|\boldsymbol{\omega}_g\|_2}, \frac{\boldsymbol{\delta}_g}{\|\boldsymbol{\delta}_g\|_2}\rangle\|\boldsymbol{\omega}_g\|_2
\end{aligned}$$

From now on, to avoid cluttering the notation assume $\mathbf{W} = \mathbf{X}_0$ and $\boldsymbol{\omega} = \boldsymbol{\omega}_0$:

$$\boldsymbol{\omega}^T\mathbf{X}\boldsymbol{\delta} = \sum_{g=0}^{G}\sqrt{\frac{n_g}{n}}\|\boldsymbol{\delta}_g\|_2\langle \mathbf{X}_g^T\frac{\boldsymbol{\omega}_g}{\|\boldsymbol{\omega}_g\|_2}, \frac{\boldsymbol{\delta}_g}{\|\boldsymbol{\delta}_g\|_2}\rangle\sqrt{\frac{n}{n_g}}\|\boldsymbol{\omega}_g\|_2$$

Assume $a_g = \sqrt{\frac{n_g}{n}}\|\boldsymbol{\delta}_g\|_2$ and $b_g = \langle \mathbf{X}_g^T\frac{\boldsymbol{\omega}_g}{\|\boldsymbol{\omega}_g\|_2}, \frac{\boldsymbol{\delta}_g}{\|\boldsymbol{\delta}_g\|_2}\rangle\sqrt{\frac{n}{n_g}}\|\boldsymbol{\omega}_g\|_2$. Then the above term is the inner product of two vectors $\mathbf{a} = (a_0, \dots, a_G)$ and $\mathbf{b} = (b_0, \dots, b_G)$ for which we have:

$$\begin{aligned}
\sup_{\mathbf{a}\in\mathcal{H}}\mathbf{a}^T\mathbf{b} &= \sup_{\|\mathbf{a}\|_1=1}\mathbf{a}^T\mathbf{b} \\
\text{(definition of the dual norm)} &\leq \|\mathbf{b}\|_\infty \\
&= \max_{g\in[G]} b_g
\end{aligned}$$

Now we can go back to the original form:

$$\begin{aligned}
\sup_{\boldsymbol{\delta}\in\mathcal{H}}\boldsymbol{\omega}^T\mathbf{X}\boldsymbol{\delta} &\leq \max_{g\in[G]}\langle \mathbf{X}_g^T\frac{\boldsymbol{\omega}_g}{\|\boldsymbol{\omega}_g\|_2}, \frac{\boldsymbol{\delta}_g}{\|\boldsymbol{\delta}_g\|_2}\rangle\sqrt{\frac{n}{n_g}}\|\boldsymbol{\omega}_g\|_2 \\
&\leq \max_{g\in[G]}\sqrt{\frac{n}{n_g}}\|\boldsymbol{\omega}_g\|_2 \sup_{\mathbf{u}_g\in\mathcal{C}_g\cap\mathbb{S}^{p-1}}\langle \mathbf{X}_g^T\frac{\boldsymbol{\omega}_g}{\|\boldsymbol{\omega}_g\|_2}, \mathbf{u}_g\rangle \qquad (3.25)
\end{aligned}$$

To avoid cluttering we name $f_g(\boldsymbol{\omega}_g, \mathbf{X}_g) = \sqrt{\frac{n}{n_g}}\|\boldsymbol{\omega}_g\|_2 \sup_{\mathbf{u}_g\in\mathcal{A}_g}\langle \mathbf{X}_g^T\frac{\boldsymbol{\omega}_g}{\|\boldsymbol{\omega}_g\|_2}, \mathbf{u}_g\rangle$ and $e_g(\tau) = \sqrt{\frac{n}{n_g}}\sqrt{(2K^2+1)n_g}\left(\upsilon_g C_g k\omega(\mathcal{A}_g) + \rho_g\sqrt{\log G} + \tau\right)$. Then from (3.25), we have:

$$\mathbb{P}\left(\sup_{\boldsymbol{\delta}\in\mathcal{H}}\boldsymbol{\omega}^T\mathbf{X}\boldsymbol{\delta} > \max_{g\in[G]} e_g(\tau)\right) \leq \mathbb{P}\left(\max_{g\in[G]} f_g(\boldsymbol{\omega}_g, \mathbf{X}_g) > \max_{g\in[G]} e_g(\tau)\right)$$

To simplify the notation, we drop arguments of f_g for now. From the union bound we have:

$$\begin{aligned}\mathbb{P}\left(\max_{g\in[G]} f_g > \max_{g\in[G]} e_g(\tau)\right) &\leq \sum_{g=0}^{G}\mathbb{P}\left(f_g > \max_{g\in[G]} e_g(\tau)\right)\\ &\leq \sum_{g=0}^{G}\mathbb{P}\left(f_g > e_g(\tau)\right)\\ &\leq G\max_{g\in[G]}\mathbb{P}\left(f_g > e_g(\tau)\right)\\ &\leq \sigma\exp\left(-\min_{g\in[G]}\left[\nu_g n_g - \log G, \frac{\tau^2}{\eta_g^2 k^2}\right]\right)\end{aligned}$$

where $\sigma = \max_{g\in[G]}\sigma_g$. ■

3.7.6 Proof of Theorem 3.17

First of all, a quick note regarding the coefficients. If we assume that all of the coefficients defined in Definition 3.15 are positive, we can reduce the constraint set of each definition to the spherical cap $\mathcal{A}_g = \mathcal{C}_g \cap \mathbb{S}^{p-1}$:

$$\begin{aligned}\rho_g(\mu_g) &= \max\left(0, \sup_{\mathbf{u},\mathbf{v}\in\mathcal{A}_g}\mathbf{v}^T\big(\mathbf{I}_g - \mu_g\mathbf{X}_g^T\mathbf{X}_g\big)\mathbf{u}\right), \quad g\in[G]\\ \xi_g(\mu_g) &= \max\left(0, \mu_g\sup_{\mathbf{v}\in\mathcal{A}_g}\mathbf{v}^T\mathbf{X}_g^T\frac{\mathbf{w}_g}{\|\mathbf{w}_g\|_2}\right), \quad g\in[G]\\ \eta_g(\mu_g) &= \max\left(0, \mu_g\sup_{\mathbf{v}\in\mathcal{A}_g,\mathbf{u}\in\mathcal{A}_0} -\mathbf{v}^T\mathbf{X}_g^T\mathbf{X}_g\mathbf{u}\right), \quad g\in[G]_\setminus\end{aligned}$$

So keeping in mind that none of the $\rho_g(\mu_g)$, $\xi_g(\mu_g)$, and $\eta_g(\mu_g)$ can be negative we work with the latter forms, i.e., where the restricted sets are spherical caps $\mathcal{A}_g$s.

We can break down the error at iteration $t+1$ to its components because of the triangle inequality $a_t = \|\boldsymbol{\delta}^{(t+1)}\|_2 \leq \sum_{g=0}^{G}\|\boldsymbol{\delta}_g^{(t+1)}\|_2 = b_t$. We analyze the convergence properties of the b_t series and the results holds automatically for a_t, since $a_t \leq b_t$. Next we upper bound the private error $\|\boldsymbol{\delta}_g^{(t+1)}\|_2$ and shared one $\|\boldsymbol{\delta}_0^{(t+1)}\|_2$ in the followings.

$$\begin{aligned}
\|\boldsymbol{\delta}_g^{(t+1)}\|_2 &= \|\boldsymbol{\beta}_g^{(t+1)} - \boldsymbol{\beta}_g^*\|_2 \\
&= \left\|\Pi_{\Omega_{R_g}}\left(\boldsymbol{\beta}_g^t + \mu_g \mathbf{X}_g^T\left(\mathbf{y}_g - \mathbf{X}_g\big(\boldsymbol{\beta}_0^t + \boldsymbol{\beta}_g^t\big)\right)\right) - \boldsymbol{\beta}_g^*\right\|_2 \\
&= \left\|\Pi_{\Omega_{R_g}-\{\boldsymbol{\beta}_g^*\}}\left(\boldsymbol{\beta}_g^t + \mu_g \mathbf{X}_g^T\left(\mathbf{y}_g - \mathbf{X}_g\big(\boldsymbol{\beta}_0^t + \boldsymbol{\beta}_g^t\big)\right) - \boldsymbol{\beta}_g^*\right)\right\|_2 \\
&= \left\|\Pi_{\mathcal{E}_g}\left(\boldsymbol{\delta}_g^t + \mu_g \mathbf{X}_g^T\left(\mathbf{y}_g - \mathbf{X}_g\big(\boldsymbol{\beta}_0^t + \boldsymbol{\beta}_g^t\big) - \mathbf{X}_g\big(\boldsymbol{\beta}_0^* + \boldsymbol{\beta}_g^*\big) + \mathbf{X}_g\big(\boldsymbol{\beta}_0^* + \boldsymbol{\beta}_g^*\big)\right)\right)\right\|_2 \\
&= \left\|\Pi_{\mathcal{E}_g}\left(\boldsymbol{\delta}_g^t + \mu_g \mathbf{X}_g^T\left(\mathbf{w}_g - \mathbf{X}_g\big(\boldsymbol{\delta}_0^t + \boldsymbol{\delta}_g^t\big)\right)\right)\right\|_2 \\
&\leq \left\|\Pi_{\mathcal{C}_g}\left(\boldsymbol{\delta}_g^t + \mu_g \mathbf{X}_g^T\left(\mathbf{w}_g - \mathbf{X}_g\big(\boldsymbol{\delta}_0^t + \boldsymbol{\delta}_g^t\big)\right)\right)\right\|_2 \\
&\leq \sup_{\mathbf{v}\in\mathcal{C}_g\cap\mathbb{B}^p} \mathbf{v}^T\left(\boldsymbol{\delta}_g^t + \mu_g \mathbf{X}_g^T\left(\mathbf{w}_g - \mathbf{X}_g\big(\boldsymbol{\delta}_0^t + \boldsymbol{\delta}_g^t\big)\right)\right) \\
&= \sup_{\mathbf{v}\in\mathcal{B}_g} \mathbf{v}^T\left(\boldsymbol{\delta}_g^t + \mu_g \mathbf{X}_g^T\left(\mathbf{w}_g - \mathbf{X}_g\big(\boldsymbol{\delta}_0^t + \boldsymbol{\delta}_g^t\big)\right)\right), \quad \mathcal{B}_g = \mathcal{C}_g \cap \mathbb{B}^p \\
&\leq \sup_{\mathbf{v}\in\mathcal{B}_g} \mathbf{v}^T\big(\mathbf{I}_g - \mu_g\mathbf{X}_g^T\mathbf{X}_g\big)\boldsymbol{\delta}_g^t + \mu_g \sup_{\mathbf{v}\in\mathcal{B}_g} \mathbf{v}^T\mathbf{X}_g^T\mathbf{w}_g + \mu_g \sup_{\mathbf{v}\in\mathcal{B}_g} -\mathbf{v}^T\mathbf{X}_g^T\mathbf{X}_g\boldsymbol{\delta}_0^t \\
&\leq \big\|\boldsymbol{\delta}_g^t\big\|_2 \sup_{\mathbf{u},\mathbf{v}\in\mathcal{B}_g} \mathbf{v}^T\big(\mathbf{I}_g - \mu_g\mathbf{X}_g^T\mathbf{X}_g\big)\mathbf{u} + \mu_g\|\mathbf{w}_g\|_2 \sup_{\mathbf{v}\in\mathcal{B}_g} \mathbf{v}^T\mathbf{X}_g^T\frac{\mathbf{w}_g}{\|\mathbf{w}_g\|_2} \\
&+ \mu_g\|\boldsymbol{\delta}_0^t\|_2 \sup_{\mathbf{v}\in\mathcal{B}_g,\mathbf{u}\in\mathcal{B}_0} -\mathbf{v}^T\mathbf{X}_g^T\mathbf{X}_g\mathbf{u} \\
&= \rho_g(\mu_g)\|\boldsymbol{\delta}_g^t\|_2 + \xi_g(\mu_g)\|\mathbf{w}_g\|_2 + \eta_g(\mu_g)\|\boldsymbol{\delta}_0^t\|_2
\end{aligned} \tag{3.26}$$

We define the following matrix:

$$\tilde{\mathbf{D}} = \begin{pmatrix} \mu_1\mathbf{X}_1 & 0 & \cdots & 0 \\ 0 & \mu_2\mathbf{X}_2 & \cdots & 0 \\ \vdots & \ddots & \cdots & \vdots \\ 0 & \cdots & \cdots & \mu_G\mathbf{X}_G \end{pmatrix} \in \mathbb{R}^{n\times Gp}$$

Then for the shared parameter have:

$$
\begin{aligned}
\|\boldsymbol{\delta}_0^{(t+1)}\|_2 &= \|\boldsymbol{\beta}_0^{(t+1)} - \boldsymbol{\beta}_0^*\|_2 \\
&= \left\| \Pi_{\Omega_{R_0}} \Big(\boldsymbol{\beta}_0^t + \mu_0 \mathbf{W}^T \Big(\mathbf{y} - \mathbf{W}\boldsymbol{\beta}_0^t - \tilde{\mathbf{D}}\boldsymbol{\beta}_{1:g}^t \Big) \Big) - \boldsymbol{\beta}_0^* \right\|_2 \\
&= \left\| \Pi_{\Omega_{R_0} - \{\boldsymbol{\beta}_0^*\}} \Big(\boldsymbol{\beta}_0^t + \mu_0 \mathbf{W}^T \Big(\mathbf{y} - \mathbf{W}\boldsymbol{\beta}_0^t - \tilde{\mathbf{D}}\boldsymbol{\beta}_{1:g}^t \Big) - \boldsymbol{\beta}_0^* \Big) \right\|_2 \\
&= \left\| \Pi_{\mathcal{E}_0} \Big(\boldsymbol{\delta}_0^t + \mu_0 \mathbf{W}^T \Big(\mathbf{y} - \mathbf{W}\boldsymbol{\beta}_0^t - \tilde{\mathbf{D}}\boldsymbol{\beta}_{1:g}^t - \mathbf{W}\boldsymbol{\beta}_0^* - \tilde{\mathbf{D}}\boldsymbol{\beta}_{1:g}^* + \mathbf{W}\boldsymbol{\beta}_0^* + \tilde{\mathbf{D}}\boldsymbol{\beta}_{1:g}^* \Big) \Big) \right\|_2 \\
&= \left\| \Pi_{\mathcal{E}_0} \Big(\boldsymbol{\delta}_0^t + \mu_0 \mathbf{W}^T \Big(\mathbf{w} - \mathbf{W}\big(\boldsymbol{\beta}_0^t - \boldsymbol{\beta}_0^*\big) - \tilde{\mathbf{D}}\big(\boldsymbol{\beta}_{1:g}^t - \boldsymbol{\beta}_{1:g}^*\big)\big) \Big) \right\|_2 \\
&= \left\| \Pi_{\mathcal{E}_0} \Big(\boldsymbol{\delta}_0^t + \mu_0 \mathbf{W}^T \Big(\mathbf{w} - \mathbf{W}\boldsymbol{\delta}_0^t - \tilde{\mathbf{D}}\boldsymbol{\delta}_{1:g}^t\big)\big) \Big) \right\|_2 \\
&\leq \left\| \Pi_{\mathcal{C}_0} \Big(\boldsymbol{\delta}_0^t + \mu_0 \mathbf{W}^T \Big(\mathbf{w} - \mathbf{W}\boldsymbol{\delta}_0^t - \tilde{\mathbf{D}}\boldsymbol{\delta}_{1:g}^t\big)\big) \Big) \right\|_2 \\
&\leq \sup_{\mathbf{v} \in \mathcal{C}_0 \cap \mathbb{B}^p} \mathbf{v}^T \Big(\boldsymbol{\delta}_0^t + \mu_0 \mathbf{W}^T \Big(\mathbf{w} - \mathbf{W}\boldsymbol{\delta}_0^t - \tilde{\mathbf{D}}\boldsymbol{\delta}_{1:g}^t\big)\big) \Big) \\
&= \sup_{\mathbf{v} \in \mathcal{B}_0} \mathbf{v}^T \Big(\boldsymbol{\delta}_0^t + \mu_0 \mathbf{W}^T \Big(\mathbf{w} - \mathbf{W}\boldsymbol{\delta}_0^t - \tilde{\mathbf{D}}\boldsymbol{\delta}_{1:g}^t\big)\big) \Big), \quad \mathcal{B}_0 = \mathcal{C}_0 \cap \mathbb{B}^p \\
&\leq \sup_{\mathbf{v} \in \mathcal{B}_0} \mathbf{v}^T \big(\mathbf{I} - \mu_0 \mathbf{W}^T\mathbf{W}\big)\boldsymbol{\delta}_0^t + \mu_0 \sup_{\mathbf{v} \in \mathcal{B}_0} \mathbf{v}^T\mathbf{W}^T\mathbf{w} + \mu_0 \sup_{\mathbf{v} \in \mathcal{B}_0} -\mathbf{v}^T\mathbf{W}^T\tilde{\mathbf{D}}\boldsymbol{\delta}_{1:g}^t \\
&\leq \sup_{\mathbf{v} \in \mathcal{B}_0} \mathbf{v}^T \big(\mathbf{I} - \mu_0 \mathbf{W}^T\mathbf{W}\big)\boldsymbol{\delta}_0^t + \mu_0 \sup_{\mathbf{v} \in \mathcal{B}_0} \mathbf{v}^T\mathbf{W}^T\mathbf{w} - \mu_0 \inf_{\mathbf{v} \in \mathcal{B}_0} \mathbf{v}^T \sum_{g=1}^{G} \mu_g \mathbf{X}_g^T\mathbf{X}_g\boldsymbol{\delta}_g^t \\
&\leq \|\boldsymbol{\delta}_0^t\|_2 \sup_{\mathbf{u},\mathbf{v} \in \mathcal{B}_0} \mathbf{v}^T \big(\mathbf{I} - \mu_0 \mathbf{W}^T\mathbf{W}\big)\mathbf{u} + \mu_0 \|\mathbf{w}\|_2 \sup_{\mathbf{v} \in \mathcal{B}_0} \mathbf{v}^T\mathbf{W}^T \frac{\mathbf{w}}{\|\mathbf{w}\|_2} \\
&+ \mu_0 \sup_{\mathbf{v} \in \mathcal{B}_0} -\mathbf{v}^T \sum_{g=1}^{G} \mu_g \mathbf{X}_g^T\mathbf{X}_g\boldsymbol{\delta}_g^t \\
&= \rho_0(\mu_0)\|\boldsymbol{\delta}_0^t\|_2 + \xi_0(\mu_0)\|\mathbf{w}\|_2 + \mu_0 \sup_{\mathbf{v} \in \mathcal{B}_0} -\mathbf{v}^T \sum_{g=1}^{G} \mu_g \mathbf{X}_g^T\mathbf{X}_g\boldsymbol{\delta}_g^t
\end{aligned}
\tag{3.27}
$$

Now we focus on the third term of (3.27):

$$\begin{aligned}
\mu_0 \sup_{\mathbf{v}\in\mathcal{A}_0} \sum_{g=1}^{G} -\mu_g \mathbf{v}^T \mathbf{X}_g^T \mathbf{X}_g \boldsymbol{\delta}_g^t &= \mu_0 \sup_{\mathbf{v}\in\mathcal{A}_0} \sum_{g=1}^{G} -\mu_g \mathbf{v}^T \mathbf{X}_g^T \mathbf{X}_g \frac{\boldsymbol{\delta}_g^t}{\|\boldsymbol{\delta}_g^t\|_2} \|\boldsymbol{\delta}_g^t\|_2 \\
&\leq \mu_0 \sum_{g=1}^{G} \mu_g \sup_{\mathbf{v}_g\in\mathcal{A}_0} -\mathbf{v}_g^T \mathbf{X}_g^T \mathbf{X}_g \frac{\boldsymbol{\delta}_g^t}{\|\boldsymbol{\delta}_g^t\|_2} \|\boldsymbol{\delta}_g^t\|_2 \\
&\leq \mu_0 \sum_{g=1}^{G} \mu_g \sup_{\mathbf{v}_g\in\mathcal{A}_0, \mathbf{u}_g\in\mathcal{A}_g} -\mathbf{v}_g^T \mathbf{X}_g^T \mathbf{X}_g \mathbf{u}_g \|\boldsymbol{\delta}_g^t\|_2 \\
&= \mu_0 \sum_{g=1}^{G} \eta_g(\mu_g) \|\boldsymbol{\delta}_g^t\|_2
\end{aligned}$$

So we rewrite the (3.27) as:

$$\|\boldsymbol{\delta}_0^{(t+1)}\|_2 \leq \rho_0(\mu_0)\|\boldsymbol{\delta}_0^t\|_2 + \xi_0(\mu_0)\|\mathbf{w}\|_2 + \mu_0 \sum_{g=1}^{G} \eta_g(\mu_g)\|\boldsymbol{\delta}_g^t\|_2$$

To avoid cluttering we drop μ_g as the arguments. Putting together (3.26) and (3.28) inequalities we reach to the followings:

$$\begin{aligned}
\|\boldsymbol{\delta}_g^{(t+1)}\|_2 &\leq \rho_g\|\boldsymbol{\delta}_g^t\|_2 + \xi_g\|\mathbf{w}_g\|_2 + \eta_g\|\boldsymbol{\delta}_0^t\|_2 \\
\|\boldsymbol{\delta}_0^{(t+1)}\|_2 &\leq \rho_0\|\boldsymbol{\delta}_0^t\|_2 + \xi_0\|\mathbf{w}\|_2 + \mu_0 \sum_{g=1}^{G} \eta_g\|\boldsymbol{\delta}_g^t\|_2
\end{aligned}$$

Also for simplicity of the notation let $\mathbf{w}_0 = \mathbf{w}$. Now we write the total error:

$$\begin{aligned}
b_t &= \sum_{g=0}^{G} \|\boldsymbol{\delta}_g^{t+1}\|_2 \\
&\leq \left(\rho_0 + \sum_{g=1}^{G} \eta_g\right) \|\boldsymbol{\delta}_0^t\|_2 + \sum_{g=1}^{G} (\rho_g + \mu_0\eta_g)\,\|\boldsymbol{\delta}_g^t\|_2 + \sum_{g=0}^{G} \xi_g\|\mathbf{w}_g\|_2
\end{aligned}$$

Let's name $\alpha = \max\left(\max_{g\in[G]_\setminus}(\rho_g + \mu_0\eta_g), \rho_0 + \sum_{g=1}^{G}\eta_g\right)$, we have:

$$
\begin{aligned}
b_t &\leq \alpha b_{t-1} + \sum_{g=0}^{G} \xi_g \|\mathbf{w}_g\|_2 \\
&\leq \alpha^2 b_{t-1} + (\alpha + 1)\sum_{g=0}^{G} \xi_g \|\mathbf{w}_g\|_2 \\
&\leq \alpha^t b_1 + \left(\sum_{i=0}^{t-1} \alpha^i\right)\sum_{g=0}^{G} \xi_g \|\mathbf{w}_g\|_2 \\
&= \alpha^t \sum_{g=0}^{G} \|\boldsymbol{\beta}_g^1 - \boldsymbol{\beta}_g^*\|_2 + \left(\sum_{i=0}^{t-1} \alpha^i\right)\sum_{g=0}^{G} \xi_g \|\mathbf{w}_g\|_2, \quad \boldsymbol{\beta}^1 = 0 \\
&\leq \alpha^t \sum_{g=0}^{G} \|\boldsymbol{\beta}_g^*\|_2 + \frac{1-\alpha^t}{1-\alpha}\sum_{g=0}^{G} \xi_g \|\mathbf{w}_g\|_2
\end{aligned}
$$

■

3.7.7 Proof of Lemma 3.18

First we upper bound each of the coefficients:

$$
\begin{aligned}
\rho_g(\mu_g) &= \sup_{\mathbf{u},\mathbf{v}\in\mathcal{B}_g} \mathbf{v}^T\big(\mathbf{I}_g - \mu_g \mathbf{X}_g^T\mathbf{X}_g\big)\mathbf{u}, \quad g \in [G] \\
&\leq 1 - \mu_g \inf_{\mathbf{u}\in\mathcal{B}_g} \|\mathbf{X}_g\mathbf{u}\|_2^2 \\
&\leq 1 - \mu_g \inf_{\mathbf{u}\in\mathcal{A}_g} \|\mathbf{X}_g\mathbf{u}\|_2^2, \quad \mathcal{A}_g = \mathcal{C}_g \cap \mathbb{S}^{p-1} \\
&= 1 - \mu_g d(\mathbf{X}_g) \\
\eta_g(\mu_g) &= \mu_g \sup_{\mathbf{v}\in\mathcal{B}_g,\mathbf{u}\in\mathcal{B}_0} -\mathbf{v}^T\mathbf{X}_g^T\mathbf{X}_g\mathbf{u} \\
&\leq \frac{\mu_g}{2} \sup_{\mathbf{v}\in\mathcal{A}_g,\mathbf{u}\in\mathcal{A}_0} \|\mathbf{X}_g\mathbf{v}\|_2^2 + \|\mathbf{X}_g\mathbf{u}\|_2^2 \\
&\leq \frac{\mu_g}{2}\left(\sup_{\mathbf{v}\in\mathcal{A}_g} \|\mathbf{X}_g\mathbf{v}\|_2^2 + \sup_{\mathbf{u}\in\mathcal{A}_0} \|\mathbf{X}_g\mathbf{u}\|_2^2\right) \\
&= \mu_g s(\mathbf{X}_g) \\
\xi_g(\mu_g) &= \mu_g \sup_{\mathbf{v}\in\mathcal{A}_g} \mathbf{v}^T\mathbf{X}_g^T \frac{\mathbf{w}_g}{\|\mathbf{w}_g\|_2}, \quad g \in [G]
\end{aligned}
$$

where $s(\mathbf{X}_g) = \frac{1}{2}\left(\sup_{\mathbf{v}\in\mathcal{A}_g}\|\mathbf{X}_g\mathbf{v}\|_2^2 + \sup_{\mathbf{u}\in\mathcal{A}_0}\|\mathbf{X}_g\mathbf{u}\|_2^2\right) \geq 0$ and $d(\mathbf{X}_g) = \inf_{\mathbf{u}\in\mathcal{A}_g}\|\mathbf{X}_g\mathbf{u}\|_2^2$. Note that $d(\mathbf{X}_g) > 0$ with high probability of $1 - 2\exp\left(-\gamma_g(\omega(\mathcal{A}_g)+\tau)^2\right)$ for enough per group number of samples, i.e., $n_g > (c_g\omega(\mathcal{A}_g) + C_g\tau)^2$.

Now writing the tail bound for each of the coefficients, starting with $\rho_g(\mu_g)$:

$$\begin{aligned}
\mathbb{P}(\rho_g(\mu_g) \geq 1 - \mu_g d_g) &\leq \mathbb{P}(1 - \mu_g d(\mathbf{X}_g) \geq 1 - \mu_g d_g) \\
&\leq \mathbb{P}(d(\mathbf{X}_g) \leq d_g) \\
\text{Lemma 3.3} &\leq 2\exp\left(-\gamma_g(\omega(\mathcal{A}_g)+\tau)^2\right)
\end{aligned}$$

For upper bound for $\eta_g(\mu_g)$ we use the law of total probability. To avoid cluttering we name $s_g = n_g + c_g\sqrt{n_g}\max(\omega(\mathcal{A}_0), \omega(\mathcal{A}_g)) + C_g\sqrt{n_g}\tau$, $\tilde{s}_g = n_g + c_g\sqrt{n_g}\omega(\mathcal{A}_g) + C_g\sqrt{n_g}\tau$ and $\tilde{s}_0 = n_g + c_g\sqrt{n_g}\omega(\mathcal{A}_0) + C_g\sqrt{n_g}\tau$.

$$\begin{aligned}
\mathbb{P}(\eta_g(\mu_g) > \mu_g s_g) &\leq \mathbb{P}(\mu_g s(\mathbf{X}_g) > \mu_g s_g) \\
&= \mathbb{P}(s(\mathbf{X}_g) > s_g) \\
&\leq \mathbb{P}(2s(\mathbf{X}_g) > \tilde{s}_0 + \tilde{s}_g) \\
&= \mathbb{P}\left(2s(\mathbf{X}_g) > \tilde{s}_0 + \tilde{s}_g \middle| \sup_{\mathbf{u}\in\mathcal{A}_g}\|\mathbf{X}_g\mathbf{u}\|_2^2 > \tilde{s}_g, \sup_{\mathbf{u}\in\mathcal{A}_0}\|\mathbf{X}_g\mathbf{u}\|_2^2 > \tilde{s}_0\right) \\
&\quad \mathbb{P}(\sup_{\mathbf{u}\in\mathcal{A}_g}\|\mathbf{X}_g\mathbf{u}\|_2^2 > \tilde{s}_g | \sup_{\mathbf{u}\in\mathcal{A}_0}\|\mathbf{X}_g\mathbf{u}\|_2^2 > \tilde{s}_0)\mathbb{P}(\sup_{\mathbf{u}\in\mathcal{A}_0}\|\mathbf{X}_g\mathbf{u}\|_2^2 > \tilde{s}_0) \\
&+ \mathbb{P}\left(2s(\mathbf{X}_g) > \tilde{s}_0 + \tilde{s}_g \middle| \sup_{\mathbf{u}\in\mathcal{A}_g}\|\mathbf{X}_g\mathbf{u}\|_2^2 > \tilde{s}_g, \sup_{\mathbf{u}\in\mathcal{A}_0}\|\mathbf{X}_g\mathbf{u}\|_2^2 < \tilde{s}_0\right) \\
&\quad \mathbb{P}(\sup_{\mathbf{u}\in\mathcal{A}_0}\|\mathbf{X}_g\mathbf{u}\|_2^2 < \tilde{s}_0 | \sup_{\mathbf{u}\in\mathcal{A}_g}\|\mathbf{X}_g\mathbf{u}\|_2^2 > \tilde{s}_g)\mathbb{P}(\sup_{\mathbf{u}\in\mathcal{A}_g}\|\mathbf{X}_g\mathbf{u}\|_2^2 > \tilde{s}_g) \\
&+ \mathbb{P}\left(2s(\mathbf{X}_g) > \tilde{s}_0 + \tilde{s}_g \middle| \sup_{\mathbf{u}\in\mathcal{A}_g}\|\mathbf{X}_g\mathbf{u}\|_2^2 < \tilde{s}_g, \sup_{\mathbf{u}\in\mathcal{A}_0}\|\mathbf{X}_g\mathbf{u}\|_2^2 > \tilde{s}_0\right) \\
&\quad \mathbb{P}(\sup_{\mathbf{u}\in\mathcal{A}_g}\|\mathbf{X}_g\mathbf{u}\|_2^2 < \tilde{s}_g | \sup_{\mathbf{u}\in\mathcal{A}_0}\|\mathbf{X}_g\mathbf{u}\|_2^2 > \tilde{s}_0)\mathbb{P}(\sup_{\mathbf{u}\in\mathcal{A}_0}\|\mathbf{X}_g\mathbf{u}\|_2^2 > \tilde{s}_0) \\
&\leq 2\mathbb{P}(\sup_{\mathbf{u}\in\mathcal{A}_0}\|\mathbf{X}_g\mathbf{u}\|_2^2 > \tilde{s}_0) + \mathbb{P}(\sup_{\mathbf{u}\in\mathcal{A}_g}\|\mathbf{X}_g\mathbf{u}\|_2^2 > \tilde{s}_g) \\
\text{Lemma 3.3} &\leq 4\exp\left(-\gamma_0(\omega(\mathcal{A}_0)+\tau)^2\right) + 2\exp\left(-\gamma_g(\omega(\mathcal{A}_g)+\tau)^2\right) \\
&\leq 4\exp\left(-\gamma\left(\min(\omega(\mathcal{A}_0), \omega(\mathcal{A}_g)) + \tau\right)^2\right)
\end{aligned}$$

where $\gamma = \min_{\gamma_g\in[G]}\gamma_g$.

From the above proof, following two inequalities will be used in future, so we separate them here:

$$\begin{aligned}\mathbb{P}(s(\mathbf{X}_g) > s_g) &\leq 4\exp\left(-\gamma\left(\min(\omega(\mathcal{A}_0), \omega(\mathcal{A}_g)) + \tau\right)^2\right) && (3.28)\\ \mathbb{P}(d(\mathbf{X}_g) < d_g) &\leq 2\exp\left(-\gamma_g(\omega(\mathcal{A}_g) + \tau)^2\right) && (3.29)\end{aligned}$$

■

3.7.8 Proof of Theorem 3.19

Throughout the proof, we assume $\forall g \in [G] : d_g > 0$ which happens if we have enough number of samples n_g (to be determined in the proof). We show that the sufficient condition for keeping $\alpha < 1$ holds with high probability, i.e., with probability at least $1-10\exp\left(-\gamma\left(\min_{g\in[G]}\omega(\mathcal{A}_g) + a\right)^2\right)$:

$$\begin{aligned}\left(\rho_0 + \sum_{g=1}^{G}\eta_g\right) &< 1\\ \forall g \in [G]_{\setminus} : (\rho_g + \mu_0\eta_g) &< 1\end{aligned}$$

Now we want to show that (3.11) condition holds with high probability. Let's simplify the condition of (3.11):

$$\begin{aligned}\rho_0 + \sum_{g=1}^{G}\eta_g &\leq 1 - \mu_0 d(\mathbf{X}_0) + \sum_{g=1}^{G}\mu_g s(\mathbf{X}_g)\\ \Rightarrow 1 - \mu_0 d(\mathbf{X}_0) + \sum_{g=1}^{G}\mu_g s(\mathbf{X}_g) &< 1\\ \Rightarrow \sum_{g=1}^{G}\mu_g s(\mathbf{X}_g) &< \mu_0 d(\mathbf{X}_0)\end{aligned}$$

We write the undesirable event probability and replace the μ_g with its Lemma's upper bound:

$$\begin{aligned}\mathbb{P}(\sum_{g=1}^{G}\mu_g s(\mathbf{X}_g) > \mu_0 d(\mathbf{X}_0)) &\leq \mathbb{P}(\sum_{g=1}^{G}\frac{\mu_0 d_0}{G s_g}s(\mathbf{X}_g) > \mu_0 d(\mathbf{X}_0))\\ &\leq \mathbb{P}(d_0\sum_{g=1}^{G}\frac{s(\mathbf{X}_g)}{s_g} > G d(\mathbf{X}_0))\end{aligned}$$

Now we write the law of total probability:

$$\begin{aligned}
& \mathbb{P}\left(d_0 \sum_{g=1}^{G} \frac{s(\mathbf{X}_g)}{s_g} > G d(\mathbf{X}_0)\right) \\
= \;& \mathbb{P}\left(d_0 \sum_{g=1}^{G} \frac{s(\mathbf{X}_g)}{s_g} > G d(\mathbf{X}_0) \Big| d_0 < d(\mathbf{X}_0)\right) \mathbb{P}(d_0 < d(\mathbf{X}_0)) \\
+ \;& \mathbb{P}\left(d_0 \sum_{g=1}^{G} \frac{s(\mathbf{X}_g)}{s_g} > G d(\mathbf{X}_0) \Big| d_0 > d(\mathbf{X}_0)\right) \mathbb{P}(d_0 > d(\mathbf{X}_0)) \\
\leq \;& \mathbb{P}\left(d_0 \sum_{g=1}^{G} \frac{s(\mathbf{X}_g)}{s_g} > G d(\mathbf{X}_0) \Big| d_0 < d(\mathbf{X}_0)\right) + \mathbb{P}(d_0 > d(\mathbf{X}_0)) \\
\leq \;& \mathbb{P}\left(\sum_{g=1}^{G} \frac{s(\mathbf{X}_g)}{s_g} > G\right) + \mathbb{P}(d_0 > d(\mathbf{X}_0)) \\
= \;& \mathbb{P}\left(\frac{s(\mathbf{X}_1)}{s_1} + \sum_{g=1}^{G-1} \frac{s(\mathbf{X}_g)}{s_g} > G \Big| s(\mathbf{X}_1) > s_1\right) \mathbb{P}(s(\mathbf{X}_1) > s_1) \\
+ \;& \mathbb{P}\left(\frac{s(\mathbf{X}_1)}{s_1} + \sum_{g=1}^{G-1} \frac{s(\mathbf{X}_g)}{s_g} > G \Big| s(\mathbf{X}_1) < s_1\right) \mathbb{P}(s(\mathbf{X}_1) < s_1) + \mathbb{P}(d_0 > d(\mathbf{X}_0)) \\
\leq \;& \mathbb{P}\left(\sum_{g=2}^{G} \frac{s(\mathbf{X}_g)}{s_g} > G - 1\right) + \mathbb{P}(s(\mathbf{X}_1) > s_1) + \mathbb{P}(d_0 > d(\mathbf{X}_0)) \\
\text{recurse} \quad \leq \;& \mathbb{P}\left(\frac{s(\mathbf{X}_G)}{s_G} > 1\right) + \sum_{g=1}^{G-1} \mathbb{P}(s(\mathbf{X}_g) > s_g) + \mathbb{P}(d_0 > d(\mathbf{X}_0)) \\
\leq \;& \sum_{g=1}^{G} \mathbb{P}(s(\mathbf{X}_g) > s_g) + \mathbb{P}(d_0 > d(\mathbf{X}_0)) \\
\leq \;& 2G \max_{g \in [G]_\setminus} \mathbb{P}(s(\mathbf{X}_g) > s_g) + \mathbb{P}(d_0 > d(\mathbf{X}_0)) \\
\leq \;& 8G \exp\left(-\gamma \min_{g \in [G]} \left(\min(\omega(\mathcal{A}_0), \omega(\mathcal{A}_g)) + \tau\right)^2\right) + \mathbb{P}(d_0 > d(\mathbf{X}_0)) \\
\leq \;& 8G \exp\left(-\gamma \left(\min_{g \in [G]} \omega(\mathcal{A}_g) + \tau\right)^2\right) + 2 \exp\left(-\gamma_g (\omega(\mathcal{A}_g) + \tau)^2\right) \\
\leq \;& 10G \exp\left(-\gamma \left(\min_{g \in [G]} \omega(\mathcal{A}_g) + \tau\right)^2\right) \\
\left(\tau = a + \sqrt{\frac{\log G}{\gamma}}\right) \quad \leq \;& 10 \exp\left(-\gamma \left(\min_{g \in [G]} \omega(\mathcal{A}_g) + a\right)^2\right) \qquad (3.30)
\end{aligned}$$

Note that (3.30) suggests $n_g \geq (\omega(\mathcal{A}_g) + C_g\sqrt{\frac{\log G}{\gamma}} + C_g a)^2$.

Similarly, let's simplify the condition of (3.12):

$$\begin{aligned}
\rho_g + \mu_0\eta_g &\leq 1 - \mu_g d(\mathbf{X}_g) + \mu_0\mu_g s(\mathbf{X}_g) \\
\Rightarrow 1 - \mu_g d(\mathbf{X}_g) + \mu_0\mu_g s(\mathbf{X}_g) &< 1 \\
\Rightarrow \mu_0\mu_g s(\mathbf{X}_g) &< \mu_g d(\mathbf{X}_g) \\
\Rightarrow \mu_0 s(\mathbf{X}_g) &< d(\mathbf{X}_g)
\end{aligned}$$

Writing the law of total probability for the event that we do not desire:

$$\begin{aligned}
\mathbb{P}(\mu_0 s(\mathbf{X}_g) > d(\mathbf{X}_g)) &= \mathbb{P}(\mu_0 s(\mathbf{X}_g) > d(\mathbf{X}_g) | d(\mathbf{X}_g) > d_g)\mathbb{P}(d(\mathbf{X}_g) > d_g) \\
&+ \mathbb{P}(\mu_0 s(\mathbf{X}_g) > d(\mathbf{X}_g) | d(\mathbf{X}_g) < d_g)\mathbb{P}(d(\mathbf{X}_g) < d_g) \\
&\leq \mathbb{P}(\mu_0 s(\mathbf{X}_g) > d_g) + \mathbb{P}(d(\mathbf{X}_g) < d_g) \\
&= \mathbb{P}(\left(\max_{i\in[G]\setminus} \frac{d_i}{s_i}\right)\frac{s_g}{d_g} s(\mathbf{X}_g) > s_g) + \mathbb{P}(d(\mathbf{X}_g) < d_g) \\
&\leq \mathbb{P}(s(\mathbf{X}_g) > s_g) + \mathbb{P}(d(\mathbf{X}_g) < d_g) \\
\text{(3.28), (3.29)} \quad &\leq 4\exp\left(-\gamma\left(\min(\omega(\mathcal{A}_0), \omega(\mathcal{A}_g)) + \tau\right)^2\right) \\
&+ 2\exp\left(-\gamma_g(\omega(\mathcal{A}_g) + \tau)^2\right) \\
&\leq 6\exp\left(-\gamma\left(\min(\omega(\mathcal{A}_0), \omega(\mathcal{A}_g)) + \tau\right)^2\right) \\
&\leq 6\exp\left(-\gamma\left(\min_{g\in[G]} \omega(\mathcal{A}_g) + \tau\right)^2\right) \\
\left(\tau = a + \sqrt{\frac{\log G}{\gamma}}\right) \quad &\leq 6\exp\left(-\gamma\left(\min_{g\in[G]} \omega^2(\mathcal{A}_g) + a^2\right)\right)
\end{aligned}$$

Finally, we want to bound the $\xi_g(\mu_g)\|\mathbf{w}_g\|_2 = \mu_g\|\mathbf{w}_g\|_2 \sup_{\mathbf{v}\in\mathcal{A}_g} \mathbf{v}^T\mathbf{X}_g^T \frac{\mathbf{w}_g}{\|\mathbf{w}_g\|_2}$. We can readily use Lemma 3.8 to get the following bound:

$$\begin{aligned}
&\mathbb{P}\left(\xi_g(\mu_g)\|\mathbf{w}_g\|_2 > \sqrt{(2K^2+1)n_g}\left(\zeta_g k\omega(\mathcal{A}_g) + \rho_g\sqrt{\log G} + \tau\right)\right) \\
\leq \quad &\frac{\sigma_g}{G}\exp\left(-\min\left[\nu_g n_g - \log G, \frac{t^2}{\eta_g^2 k^2}\right]\right)
\end{aligned}$$

So with probability $1 - \sigma_g \exp\left(- \min\left[\nu_g n_g - \log G, \frac{t^2}{\eta_g^2 k^2}\right]\right)$ we have:

$$\begin{aligned}
\sum_{g=1}^{G} \xi_g(\mu_g) \|\mathbf{w}_g\|_2 &\leq \sqrt{(2K^2+1)} \sum_{g=1}^{G} \mu_g \sqrt{n_g} \left(\zeta_g k \omega(\mathcal{A}_g) + \rho_g \sqrt{\log G} + \tau\right) \\
(\mu_g \leq \frac{1}{G}) &\leq \sqrt{(2K^2+1)n} \left(\zeta k \max_{g \in [G]} \omega(\mathcal{A}_g) + \rho \sqrt{\log G} + \tau\right)
\end{aligned}$$

Chapter 4

Structured Regression with Noisy Covariates

Error in features is known with different names in the literature such as measurement error, errors-in-variables, or noisy covariates, and has applications in various areas of science and engineering [76, 29, 50]. The importance of measurement error models is amplified in the era of big data, since large scale and high dimensional data are more prone to noise [22, 50]. In high dimensional setting where $p \gg n$ the classical assumptions required for treatment of measurement error break down [76, 29] and new estimators and methods are required to consistently estimate $\boldsymbol{\beta}^*$. Such challenges have revived measurement error research and several papers have addressed high dimensional issues of those models in recent years [51, 49, 22, 50, 52].

Many recent papers have reported unstable behavior of standard sparse estimators like LASSO [9] and Dantzig selector (DS) [11] under measurement error. These observations, led to suggestion of new estimators [51, 49, 22, 50, 52] for which some knowledge of noise $\mathbf{w}_i$, and/or $\boldsymbol{\beta}^*$ are required for consistent estimation. None of the existing estimators is able to consistently estimate parameters from noisy measurements without noise information, but there is still no theoretical result to show inachievability.

Here, we consider regularized (LASSO type) estimators with general norms $R(\cdot)$, when the design matrix $\mathbf{X}$, with $\mathbf{x}_i$ as its rows, is corrupted by additive independent sub-Gaussian noise matrix $\mathbf{W}$ (precise definition of sub-Gaussian random variable follows). Therefore, the additive noise model in matrix form becomes:

$$\begin{aligned} \mathbf{Z} &= \mathbf{X} + \mathbf{W}, \quad \mathbf{Z}, \mathbf{X}, \mathbf{W} \in \mathbb{R}^{n \times p} \qquad (4.1) \\ \mathbf{y} &= \mathbf{X}\boldsymbol{\beta}^* + \boldsymbol{\epsilon}, \quad \mathbf{y}, \boldsymbol{\beta}, \boldsymbol{\epsilon} \in \mathbb{R}^p, \end{aligned}$$

where matrix $\mathbf{Z}$ is the noisy observation (design) matrix with $\mathbf{z}_i$s as its rows which follow additive noise model of (1.6) and $\mathbf{y}$ is generated from linear model of (1.5). Our regularized estimator takes the form:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathcal{C}}{\operatorname{argmin}} \, \mathcal{L}(\mathbf{Z}, \mathbf{y}, \boldsymbol{\beta}) + \lambda R(\boldsymbol{\beta}), \qquad (4.2)$$

where $\mathcal{L}$ is a loss function, $\mathcal{C} \subseteq \mathbb{R}^p$ and $R(\cdot)$ is a general norm used for regularization and induces some structure (like sparsity) over the unknown parameter $\boldsymbol{\beta}^*$.

To the best of our knowledge none of the previous work in high dimensional measurement error literature (see Section 2.2 on the related work) has considered structures other than sparsity, i.e. $R(\boldsymbol{\beta}^*) = \|\boldsymbol{\beta}^*\|_1$. However, other structures of $\boldsymbol{\beta}^*$ are of interest in different applications [48, 14, 42, 12]. These structures are formalized as having a small value for $R(\boldsymbol{\beta}^*)$ where R is a suitable norm.

In this chapter, we first study the properties of the estimator (4.2) where no knowledge of the noise W is available. This is in the sharp contrast to the recent literature [49, 22, 50] where the noise covariance $\Sigma_{\mathbf{w}} = \mathbb{E}[\mathbf{W}^T\mathbf{W}] \in \mathbb{R}^{p \times p}$ or an estimate of it, is required as a part of estimator. [22] uses a maximum likelihood estimator, which always requires estimation of $\Sigma_{\mathbf{w}}$ in order to establish restricted eigenvalue conditions [69, 47, 53] on the estimated sample covariance $\Sigma_{\mathbf{x}}$. [49] used orthogonal matching pursuit to recover the support of $\boldsymbol{\beta}^*$ without any knowledge of $\Sigma_{\mathbf{w}}$, but it can not establish l_2 consistency without estimating $\Sigma_{\mathbf{w}}$ directly. Our analysis of estimator (4.2) when $\Sigma_{\mathbf{w}}$ is unknown characterizes the upper bound on $\|\boldsymbol{\delta}\|_2 \leq g(n) + c(\Sigma_{\mathbf{w}})$, where $g(n)$ decays by the rate of $O(1/\sqrt{n})$ but the constant $c(\Sigma_{\mathbf{w}})$, is not vanishing. Thus, the upper bound on the statistical error does not decay to zero, but remains bounded within a norm ball. Second, we prove that when $\Sigma_{\mathbf{w}}$ is available, the regularized estimators like (4.2) are consistent which generalizes the recent work of [22] for the case of $R(\cdot) = \|\cdot\|_1$.

The rest of the chapter is organized as follows. First, in Section 4.1 we formulate the structured estimation problem under noisy designs assumption using regularized optimization and establish non-asymptotic bounds on the error for sub-Gaussian designs and sub-Gaussian noise.

In Section 4.2, we prove consistency of estimators when an estimate $\hat{\Sigma}_{\mathbf{w}}$ of noise covariance is known. We present supportive numerical simulation results in Section 4.3.

4.1 Statistical Properties

We consider the linear model, where covariates are corrupted by additive noise $y_i = \langle \mathbf{x}_i, \boldsymbol{\beta}^* \rangle + \epsilon_i, \quad \mathbf{z}_i = \mathbf{x}_i + \mathbf{w}_i$, where $\mathbf{x}_i \sim \text{Subg}(0, \Sigma_{\mathbf{x}}, K_{\mathbf{x}})$, $\epsilon_i \sim \text{Subg}(0, \sigma_\epsilon, K_\epsilon)$ are i.i.d and also independent from one another. Error vector $\mathbf{w}_i \sim \text{Subg}(0, \Sigma_{\mathbf{w}}, K_{\mathbf{w}})$ is independent from both $\mathbf{x}_i$ and ϵ_i. Since $\mathbf{w}_i$ and $\mathbf{x}_i$ are independent, we have $\Sigma_{\mathbf{z}} = \Sigma_{\mathbf{x}} + \Sigma_{\mathbf{w}}$ and $\mathbf{z}_i \sim \text{Subg}(0, \Sigma_{\mathbf{z}}, K_{\mathbf{z}})$ for $K_{\mathbf{z}} \leq c_1 K_{\mathbf{x}} + c_2 K_{\mathbf{w}}$. In matrix notation, given samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, we obtain

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta}^* + \boldsymbol{\epsilon}, \quad \mathbf{Z} = \mathbf{X} + \mathbf{W} \,. \tag{4.3}$$

The regularized family of estimators in high dimensions is generally characterized as

$$\hat{\boldsymbol{\beta}}_r = \underset{\boldsymbol{\beta}}{\text{argmin}} \frac{1}{2n} \|\mathbf{y} - \mathbf{Z}\boldsymbol{\beta}\|_2^2 + \lambda_r R(\boldsymbol{\beta}), \tag{4.4}$$

where $\lambda_r > 0$.

In noiseless scenario, i.e. $\mathbf{Z} = \mathbf{X}$, (4.4) is called Regularized M-estimators (RME) [14, 12]. R encodes the structure of $\boldsymbol{\beta}^*$. For example, if $\boldsymbol{\beta}^*$ is sparse, i.e. has many zeros, $R(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_1$ and RME (4.4) corresponds to the LASSO problem [9]. When $\mathbf{Z} = \mathbf{X}$, statistical consistency of RME has been shown for general norms [14].

For noiseless designs, considerable progress has been made in recent years in the analysis of non-asymptotic estimation error $\|\boldsymbol{\delta}\|_2 = \|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2$ [14, 77, 57, 12, 78]. In this paper, we follow the established analysis techniques, while discussing some of the subtle differences in the results obtained due to presence of the noise in covariates. First we discuss the set of directions which contain the error $\boldsymbol{\delta}$.

Lemma 4.1 (Error Set [14]) *Choosing $\lambda_r \geq \alpha R^*(\frac{1}{n}\mathbf{Z}^T(\mathbf{y} - \mathbf{Z}\boldsymbol{\beta}^*))$ for some $\alpha > 1$, the error vector $\boldsymbol{\delta}$ of RME (4.4) belongs to the restricted error set E_r [14]*

$$\mathcal{E}_r = \left\{ \boldsymbol{\delta} \in \mathbb{R}^p \middle| R(\boldsymbol{\beta}^* + \boldsymbol{\delta}) \leq R(\boldsymbol{\beta}^*) + \frac{1}{\alpha} R(\boldsymbol{\delta}) \right\} \tag{4.5}$$

We name the cone of $\mathcal{E}_r$ as $\mathcal{C}_r = Cone(\mathcal{E}_r)$.

Proof is straightforward and similar to [14], which only depends on the optimality of $\hat{\boldsymbol{\beta}}$. Next, we discuss the Restricted Eigenvalue (RE) condition on the design matrix that almost all of the high-dimensional consistency analysis relies on [14, 42, 57, 22, 12, 50, 52].

Definition 4.2 (Restricted Eigenvalue) *The design matrix $\mathbf{Z} \in \mathbb{R}^{n \times p}$ satisfies the restricted eigenvalue condition on the spherical cap $\mathcal{A} \subset \mathbb{S}^{p-1}$, where $\mathbb{S}^{p-1}$ is the unit l_2 sphere, if $\frac{1}{\sqrt{n}} \inf_{\mathbf{v} \in \mathcal{A}} \|\mathbf{X}\mathbf{v}\|_2 \geq \kappa > 0$ or in other words, for $\gamma = \sqrt{n}\kappa$:*

$$\inf_{\mathbf{v} \in \mathcal{A}} \|\mathbf{X}\mathbf{v}\|_2 \geq \gamma > 0 \, . \tag{4.6}$$

Intuitively RE condition means that although for $p \gg n$ the matrix X is not positive definite and the corresponding quadratic form is not strongly convex but in the certain desirable directions represented by $\mathcal{A}$, $||\mathbf{X}\mathbf{v}\|_2^2$ is strongly convex. In RME these are error vector $\boldsymbol{\delta}$ directions formulated as $\mathcal{A}_r = \mathcal{C}_r \cap \mathbb{S}^{p-1}$.

For the noiseless case $\mathbf{Z} = \mathbf{X}$ when $\mathbf{x}_i$ are Gaussian or sub-Gaussian RE condition is satisfied with high probability after a certain sample size $n > n_0$ is reached, where n_0 determines the sample complexity [14, 12]. Interestingly, recent work has shown that the sample complexity is the square of the Gaussian width of $\mathcal{A}$, $n_0 = O(\omega^2(\mathcal{A}))$ [14].

Theorem 4.3 (Deterministic Error Bound [14, 57]) *Assume $\lambda_r \geq \alpha R^*(\frac{1}{n}\mathbf{Z}^T(\mathbf{y} - \mathbf{Z}\boldsymbol{\beta}^*))$ for some $\alpha > 1$ and sample size $n > n_0$ such that RE condition (4.6) holds over the error directions $\mathcal{A}_r = \mathcal{C}_r \cap \mathbb{S}^{p-1}$, then following deterministic bound holds for RME:*

$$\|\boldsymbol{\delta}_r\|_2 \leq \frac{\alpha + 1}{\alpha} \frac{\lambda_r}{\kappa} \Psi(\mathcal{C}_r) \, , \tag{4.7}$$

where $\Psi(\mathcal{C}) = \sup_{\mathbf{u} \in \mathcal{C}} \frac{R(\mathbf{u})}{\|\mathbf{u}\|_2}$ is the restricted norm compatibility constant.

Next, we analyze the additive noise case, by (i) obtaining suitable bounds for λ, which sets the scaling of the error bound, and (ii) the sample complexity n_0 for which the RE condition is satisfied with high-probability even with a noisy design $\mathbf{Z}$. Without loss of generality, we will assume $\|\boldsymbol{\beta}^*\|_2 = 1$ for the analysis, noting that the general case follows by a direct scaling of the analysis presented.

4.1.1 Restricted Eigenvalue Condition

For linear models with the square loss function, RE condition is satisfied if (4.6) holds, where $\mathcal{A} \subseteq \mathbb{S}^{p-1}$ is a restricted set of directions. Recent literature [14, 42, 12] has proved that the RE condition holds for both Gaussian and sub-Gaussian design matrices. In the following theorem we show that RE condition holds for additive noise in measurement with high probability:

Theorem 4.4 *For the design matrix of the additive noise in measurement* $\mathbf{Z} = \mathbf{X} + \mathbf{W}$ *where independent rows of* $\mathbf{X}$ *and* $\mathbf{W}$ *are drawn from* $\mathbf{x}_i \sim$ *Subg*$(0, \Sigma_{\mathbf{x}}, K_{\mathbf{x}})$, *and* $\mathbf{w}_i \sim$ *Subg*$(0, \Sigma_{\mathbf{w}}, K_{\mathbf{w}})$, *for absolute constants* $\eta, c > 0$, *with probability at least* $(1 - 2\exp(-\eta\omega^2(\mathcal{A})))$, *we have:*

$$\inf_{\mathbf{v} \in \mathcal{A}} \frac{1}{n} \|\mathbf{Z}\mathbf{v}\|_2^2 \geq \lambda_{\min}(\Sigma_{\mathbf{x}} + \Sigma_{\mathbf{w}} | \mathcal{A}) \left(1 - c\frac{\omega(\mathcal{A})}{\sqrt{n}}\right) , \tag{4.8}$$

where $\mathcal{A} \subseteq \mathbb{S}^{p-1}$.

Proof: Note that $\mathbf{Z} = \mathbf{X} + \mathbf{W}$ and since rows of $\mathbf{X}$ and $\mathbf{W}$ are centered independent and sub-Gaussian, as mentioned in Section 4.1 rows of $\mathbf{Z}$ are also sub-Gaussian following distribution $\mathbf{z}_i \sim \text{Subg}(0, \Sigma_{\mathbf{x}} + \Sigma_{\mathbf{w}}, cK_{\mathbf{x}} + CK_{\mathbf{w}})$. Now we apply Theorem 10 of [14] for RE condition of independent anisotropic sub-Gaussian designs and result follows. ■

In the noisy design problem, our quantity of interest is the Gaussian width $\omega(\mathcal{A}_r)$. For example, l_1 norm in LASSO is a simple special case of this model where $\boldsymbol{\beta}^*$ is s-sparse and we obtain $\omega(\mathcal{A}) \leq \sqrt{s \log p}$ [14, 42]. Further, Group-LASSO is the generalization of LASSO to group-sparse norms, where one considers that the dimensions $1, \ldots, p$ are grouped into n_G disjoint groups each of size at most m_G, and $\boldsymbol{\beta}^*$ consists of s_G groups. In this scenario, one obtains $\omega(\mathcal{A}) \leq \sqrt{m_G} + \sqrt{s_G \log n_G}$ [56, 79]. The k-support norm was introduced in [48] and [57] provided recovery guarantees for k-support norm for linear models. It was shown in [57] that the Gaussian width of the unit ball of the k-support norm is bounded as $\omega(\Omega_{\|\cdot\|_k^{sp}}) \leq \left(\sqrt{2k \log\left(\frac{pe}{k}\right)} + \sqrt{k}\right)$. For related results we refer the readers to [80].

4.1.2 Regularization Parameter

The statistical analysis of RME requires $\lambda \geq \alpha R^*(\frac{1}{n}\mathbf{Z}^T(\mathbf{y} - \mathbf{Z}\boldsymbol{\beta}^*))$. For the noiseless case, we note that $\mathbf{y} - \mathbf{Z}\boldsymbol{\beta}^* = \mathbf{y} - \mathbf{X}\boldsymbol{\beta}^* = \boldsymbol{\epsilon}$, the noise vector, so that $R^*(\frac{1}{n}\mathbf{Z}^T(\mathbf{y} - \mathbf{Z}\beta^*)) = R^*(\frac{1}{n}\mathbf{X}^T\boldsymbol{\epsilon})$. Using the fact that X and $\boldsymbol{\epsilon}$ are sub-Gaussian and independent, recent work has shown that $E[R^*(\frac{1}{n}\mathbf{X}^T\boldsymbol{\epsilon})] \leq \frac{c}{\sqrt{n}}\omega(\Omega_R)$, where $\Omega_R = \{\mathbf{u} \in \mathbf{R}^p | R(\mathbf{u}) \leq 1\}$. For l_1 norm, i.e., LASSO, Ω_R is the unit l_1 ball, and $\omega(\Omega_R) \leq c_2\sqrt{\log p}$. Here we have the following bound on λ:

Theorem 4.5 *Assume that* $\mathbf{X}$ *and* $\mathbf{W}$ *are matrices with iid rows drawn from zero mean sub-Gaussian distributions. Then,*

$$\mathbb{E}\left[R^*\left(\frac{1}{n}\mathbf{Z}^T(\mathbf{y} - \mathbf{Z}\boldsymbol{\beta}^*)\right)\right] \leq \nu R(\boldsymbol{\beta}^*) + \frac{C\omega(\Omega_R)}{\sqrt{n}} , \tag{4.9}$$

where $\nu = \sup_{\mathbf{u}\in\Omega_R} \|\Sigma_{\mathbf{w}}^{1/2}\mathbf{u}\|_2^2$*, and* $C > 0$ *is a constant dependent on the sub-Gaussian norms of the* $\mathbf{X}$ *and* $\mathbf{W}$*.*

Remark 4.6 *For the intuitive interpretation of* (4.31)*, note that when the number of samples* n *increases sample covariance converges as* $\frac{1}{n}\mathbf{W}^T\mathbf{W} \to \Sigma_{\mathbf{w}} = I$*, therefore* $\mathbb{E}\left[R^*\left(\frac{1}{n}\mathbf{W}^T\mathbf{W}\boldsymbol{\beta}^*\right)\right] = R^*\left(\boldsymbol{\beta}^*\right)$ *which is not decaying by number of samples. Moreover,* $R^*\left(\boldsymbol{\beta}^*\right) = \sup_{\mathbf{u}\neq 0} \frac{\langle \boldsymbol{\beta}^*, \mathbf{u}\rangle}{R(\mathbf{u})} = R(\boldsymbol{\beta}^*) \sup_{\mathbf{u}\neq 0} \frac{\langle \boldsymbol{\beta}^*/R(\boldsymbol{\beta}^*), \mathbf{u}\rangle}{R(\mathbf{u})} = R(\boldsymbol{\beta}^*) \sup_{\mathbf{u}\in\Omega_R} \|\mathbf{u}\|_2^2$ *which is exactly RHS when* $n \to \infty$*.*

Remark 4.7 *Theorem 4.5 illustrates that* λ *does not decay to 0 with increasing sample size, but approaches the operator norm of the covariance matrix* $\Sigma_{\mathbf{w}}$*. Particularly, when the noise* $\mathbf{W}$ *is i.i.d. with variance* σ_w^2*, the error is bounded above by* σ_w^2*.*

Remark 4.8 *The main consequence of Theorem 4.5 is to illustrate that the existing technique for proving consistency for the statistical error* $\|\boldsymbol{\delta}\|_2$ *of the noiseless estimator fails for RME. We note that in* (4.7)*, when* $n > n_0$*,* κ *is a positive quantity that approaches the minimum eigenvalue of* $\Sigma_{\mathbf{x}} + \Sigma_{\mathbf{w}}$ *with increasing sample size. Therefore, the scaling of* λ *determines the error bounds. Theorem 4.5 proves that the error bound can be as small as the variance of the noise. When* $\mathbf{W} = 0$*, consistency rates are exactly the same as the noiseless case [14].*

4.2 Consistency With Noise Covariance Estimates

Theorem 4.5 shows that with no informations about the noise, current analyses can not guarantee statistical consistency for noisy covariates model. At the same time, appearance of $\Sigma_{\mathbf{w}}$ in the upper bound of (4.9), suggests the use of noise covariance estimate to make the estimators consistent. Motivated by this observation and recent line of work [22, 81], we focused on scenarios in which an estimate of the noise covariance matrix $\hat{\Sigma}_{\mathbf{w}}$ is available, e.g., from repeated measurements $\mathbf{Z}$ for the same design matrix $\mathbf{X}$, or from independent samples of $\mathbf{W}$. We follow [22] and assume that independent observation from zero mean noise matrix $\mathbf{W}$ is possible, from which we estimate the sample covariance as $\hat{\Sigma}_{\mathbf{w}} = \frac{1}{n}\mathbf{W}_0^T\mathbf{W}_0$. Having $\hat{\Sigma}_{\mathbf{w}}$ in hand we modify RME in the following way. Consider the matrix $\hat{\Gamma} = \frac{1}{n}\mathbf{Z}^T\mathbf{Z} - \hat{\Sigma}_{\mathbf{w}}$ where $\hat{\Sigma}_{\mathbf{w}}$ compensates the effect of noise $\mathbf{W}$, then:

$$\text{Noisy RME: } \hat{\boldsymbol{\beta}}_r = \underset{R(\boldsymbol{\beta})\leq b}{\operatorname{argmin}} \, \boldsymbol{\beta}^T\hat{\Gamma}\boldsymbol{\beta} - \boldsymbol{\beta}^T\frac{1}{n}\mathbf{Z}^T\mathbf{y} + \lambda R(\boldsymbol{\beta}) \, , \tag{4.10}$$

Program (4.10) can be non-convex, because $\hat{\Gamma} = \frac{1}{n}\mathbf{Z}^T\mathbf{Z} - \hat{\Sigma}_{\mathbf{w}}$ may be indefinite. In such a situation the objective is unbounded below. So we need to impose further constraint of the form $R(\boldsymbol{\beta}) \leq b$ where for the feasibility of $\boldsymbol{\beta}^*$ we set $b = R(\boldsymbol{\beta}^*)$. Our consistency guarantee considers the global solution $\hat{\boldsymbol{\beta}}_r$ of the non-convex problem (4.10). The relation between global and local solutions has been investigated in [22] for the special case of l_1 norm, and for general norms we leave it for the future work. Note that (4.10) "extends" estimator of [22] for any norm, i.e., for $R(\cdot) = \|\cdot\|_1$, (4.10) reduces to the objective of [22].

To show the statistical consistency of $\hat{\boldsymbol{\beta}}$ of noisy RME (NRME), similar to the noiseless case, we need three ingredients, i.e., restricted error set, bound on regularization parameter, and RE condition. The restricted error set of NRME is determined by feasibility of $\hat{\boldsymbol{\beta}}$ as follows:

$$\mathcal{E}_w \quad = \quad \left\{ \boldsymbol{\delta} \in \mathbb{R}^p \middle| R(\boldsymbol{\beta}^* + \boldsymbol{\delta}) \leq R(\boldsymbol{\beta}^*) \right\} \tag{4.11}$$

Note that the restricted error set of the noisy case is a subset of that of noiseless case, i.e., $\mathcal{E}_w \subseteq \mathcal{E}_r$. Following lemmas shows bounds on λ and RE condition for NRME.

Lemma 4.9 (Bound on λ for NRME) *With probability* $1 - c_1 \exp\{-\min(c_2\tau^2, c_3 n)\}$*:*

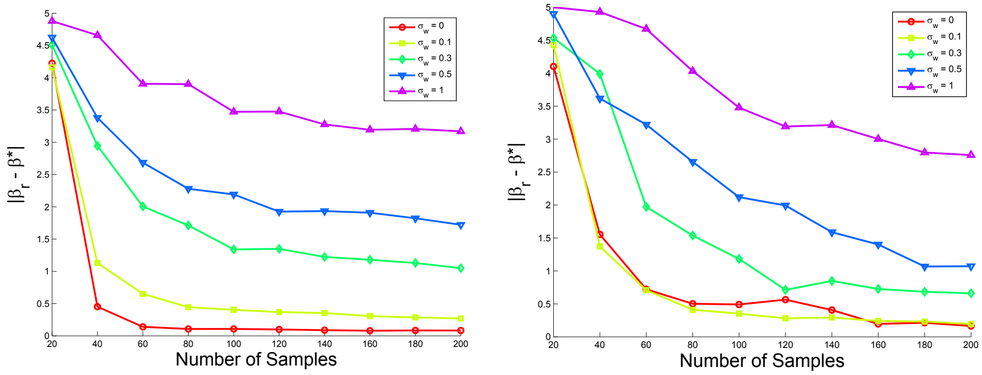
$$R^*\left(\frac{1}{n}\mathbf{Z}^T\mathbf{y} - \hat{\Gamma}\boldsymbol{\beta}^*\right) \leq \frac{c\omega(\Omega_R) + C\tau}{\sqrt{n}}. \tag{4.12}$$

Lemma 4.10 (RE condition for NRME) *For matrix* $\hat{\Gamma} = \frac{1}{n}\mathbf{Z}^T\mathbf{Z} - \hat{\Sigma}_{\mathbf{w}}$ *in the NRME objective with* $\mathbf{Z} = \mathbf{X} + \mathbf{W}$ *where independent rows of* $\mathbf{X}$ *and* $\mathbf{W}$ *are drawn from* $\mathbf{x}_i \sim$ *Subg*$(0, \Sigma_{\mathbf{x}}, K_{\mathbf{x}})$*, and* $\mathbf{w}_i \sim$ *Subg*$(0, \Sigma_{\mathbf{w}}, K_{\mathbf{w}})$*, and* $\hat{\Sigma}_{\mathbf{w}} = \frac{1}{n}\mathbf{W}_0^T\mathbf{W}_0$*, for absolute constants* $\eta, c_i > 0$*, with probability at least* $(1 - 2\exp(-\eta\omega^2(\mathcal{A}_w)))$*, we have:*

$$\begin{aligned} & \inf_{\mathbf{v} \in \mathcal{A}_w} \mathbf{v}^T\hat{\Gamma}\mathbf{v} \\ \geq \quad & \lambda_{\min}(\Sigma_{\mathbf{x}}|\mathcal{A}_w)\left(1 - c_1\frac{\omega(\mathcal{A}_w)}{\sqrt{n}}\right) \\ - \quad & c_2(\lambda_{\min}(\Sigma_{\mathbf{w}}|\mathcal{A}_w) + \lambda_{\max}(\Sigma_{\mathbf{w}}|\mathcal{A}_w))\frac{\omega(\mathcal{A}_w)}{\sqrt{n}}\ , \end{aligned} \tag{4.13}$$

where $\mathcal{A}_w \subseteq$ *Cone*$(\mathcal{E}_w) \cap \mathbb{S}^{p-1}$*.*

Note that if we set $\Sigma_{\mathbf{w}} = 0$ in (4.13) we get the established RE condition of the noiseless case [14].



(a) LASSO

(b) Noisy RME

Figure 4.1: l_2 error vs. number of samples n.

Corollary 4.11 *When number of samples n passes $n_0 = O(\omega^2(\mathcal{A}_w))$, the objective of NRME (4.10) becomes strongly convex in the direction of restricted error set $\mathcal{E}_w$.*

The following theorem shows that NRME (4.10) consistently estimates $\boldsymbol{\beta}^*$.

Theorem 4.12 *For the design matrix of the additive noise in measurement $\mathbf{Z} = \mathbf{X} + \mathbf{W}$ where independent rows of $\mathbf{X}$ and $\mathbf{W}$ are drawn from $\mathbf{x}_i \sim Subg(0, \Sigma_{\mathbf{x}}, K_{\mathbf{x}})$, and $\mathbf{w}_i \sim Subg(0, \Sigma_{\mathbf{w}}, K_{\mathbf{w}})$, and for the noise covariance estimate $\hat{\Sigma}_{\mathbf{w}} = \frac{1}{n}\mathbf{W}_0^T\mathbf{W}_0$ discussed above we have the following error bound for regularized estimator (4.10):*

$$\|\boldsymbol{\delta}\|_2 \leq \frac{2c\Psi(C_r)}{\kappa}\frac{\omega(\Omega_R)}{\sqrt{n}} \ , \tag{4.14}$$

with probability greater than $(1 - c_3\exp(-c_4\omega^2(\mathcal{A}_w)))$, where $c_3, c_4 > 0$ are constants.

Remark 4.13 *Note that when R is the vector l_1-norm $\omega(\Omega_R) \leq \sqrt{s\log p}$, and we get the rate of $O(\sqrt{\frac{s\log p}{n}})$ for* (4.14) *which matches the NCL bound of [22]. Note that the NCL [22] bound hinges on the decomposability of the l_1 norm regularizer. Our analysis for* (4.14) *does not assume decomposability, and follow arguments developed in [57].*

4.3 Experiments

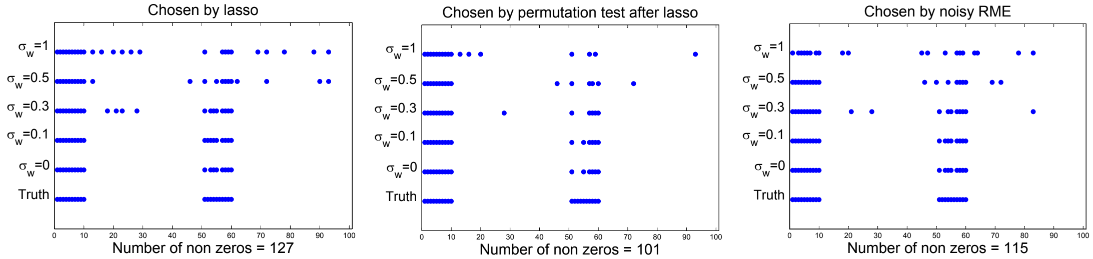
In this section we provide numerical simulations to confirm our theoretical results of Section 4.1. We focus on sparse recovery using noisy RME, i.e., $R(\boldsymbol{\beta}) = ||\boldsymbol{\beta}||_1$ and investigate l_2-norm consistency.

4.3.1 l_2 Error Bound

Experiments with l_2 norm consistency involves observing the norm of the error $\|\Delta\|_2$ which theory predicts it should decrease with the rate of $\frac{1}{\sqrt{n}}$ and converge to some positive number depending on $\Sigma_{\mathbf{w}}$. We generate synthetic data from the model of Section 4.1 with $\boldsymbol{\beta}^* = (\overbrace{-2,-2,\dots,-2}^{s/2},\overbrace{1,1,\dots,1}^{s/2},\overbrace{0,\dots,0}^{p-s})$, $\mathbf{x}_i \sim N(0, \mathbf{I}_{p\times p})$, $\mathbf{w}_i \sim N(0, \sigma_w^2 \mathbf{I}_{p\times p})$, and $\epsilon_i \sim N(0, 0.1)$ where $p = 100$, $\sigma_w^2 \in \{0, 0.1, 0.3, 0.5, 1\}$ and $s = 10$. Note that setting $\sigma_w^2 = 0$ results in the standard noiseless linear model. Figure 4.1 shows that $\|\hat{\boldsymbol{\beta}}_r - \boldsymbol{\beta}\|_2$ decreases with increasing number of samples. Each point is an average of 50 runs of the experiment. Clearly, when we increase the noise variance σ_w^2, LASSO is unable to recover the true parameter vector: with 200 samples in noiseless case error drops to $\|\boldsymbol{\delta}\|_2 \simeq 0.08$ while with noise of $\sigma_w = 1$ it stays around 3. Next we use the Noisy RME estimator and depict the same diagram in Figure 4.1b. In all level of noise, $\|\boldsymbol{\delta}\|_2$ error drops with the similar rate and with 200 samples converges to smaller value than the original estimator.

4.3.2 Noisy RME vs. Stable Feature Selection

Different level of noise in the covariates will effect the features being picked by LASSO. We perform significance test and show that in the case of noisy covariates it is helpful in recovering the true support of the parameter vector. The major problem with significance testing is that, first, one should solve the estimation problem, e.g., LASSO, several times which is not desirable. Secondly, if LASSO de-selects a feature in first place there is no chance that permutation



(a) Features selected by LASSO

(b) LASSO followed by permutation test.

(c) Features selected by NRME

Figure 4.2: Comparison between stability of LASSO, LASSO + significance test, and NRME.

test can pick it up. We show that Noisy RME can be a suitable replacement for LASSO followed by significance testing.

We pick permutation test [82, 83] as our significance testing method. In permutation test we randomly shuffle the output variables y for $v = 1000$ times and each time perform the estimation using LASSO on $\{(\mathbf{x}_i, \pi(y_i))\}_{i=1}^n$ where π is the permutation function. Name the output of LASSO on each permuted data set as $\tilde{\boldsymbol{\beta}}$ and the output of the LASSO on original samples as $\hat{\boldsymbol{\beta}}$. Then we compute the following probability:

$$p_i = \frac{\text{count}(|\tilde{\beta}_i| \geq |\hat{\beta}_i|)}{v+1} \tag{4.15}$$

For $\hat{\beta}_i$ to be a significance coefficient, p_i should be greater than 0.05. We call those $\hat{\beta}_i$s significance factors. For this experiment we set $\boldsymbol{\beta}^* = (\overbrace{-2, -2, \ldots, -2}^{1-10}, 0, \ldots, 0, \overbrace{1, 1, \ldots, 1}^{51-60}, 0, \ldots, 0)$.

Figure 4.2 show the result of stability experiment. Each row of diagrams represent the sparsity pattern (i.e., support) of the estimated vector $\hat{\boldsymbol{\beta}}$ except the lowest row which represent the sparsity pattern of true parameter vector $\boldsymbol{\beta}^*$. Figure 4.2a illustrates the features picked by LASSO. As we expect when the noise level increases LASSO starts selecting incorrect support and missing the correct support. To avoid this we perform permutation test after LASSO and get the 4.2b which clearly conforms more to the support of β^*. Although permutation test removes most of the non-support features, at the same time it discards some support feature for even small amount of noise. In contrast noisy RME of 4.2c consistently selects most part of support even for $\sigma_w = 1$. As we expect number of nonzero elements (selected features) by permutation test (101) is less than features selected by LASSO (127), since significance test only select important subset of picked features. Note that number of features picked by noisy RME (115) is the closest (on average) to actual number of support ($120 = 6 \times 20$).

4.4 Proofs

4.4.1 Proof of Theorem 4.5

Proof: Noting $\mathbf{Z} = \mathbf{X} + \mathbf{W}$ we can see that

$$\mathbf{Z}^T(\mathbf{y} - \mathbf{Z}\boldsymbol{\beta}^*) = \mathbf{Z}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^* - \mathbf{W}\boldsymbol{\beta}^*) = \mathbf{Z}^T\boldsymbol{\epsilon} - \mathbf{Z}^T\mathbf{W}\boldsymbol{\beta}^* \,. \tag{4.16}$$

Note that there is an additional term $Z^T W \boldsymbol{\beta}^*$ as a consequence of the noise. Now, by triangle inequality

$$R^*(\frac{1}{n}\mathbf{Z}^T(\mathbf{y}-\mathbf{Z}\boldsymbol{\beta}^*)) \leq R^*(\frac{1}{n}\mathbf{Z}^T\boldsymbol{\epsilon}) + R^*(\frac{1}{n}\mathbf{Z}^T\mathbf{W}\boldsymbol{\beta}^*) \; . \tag{4.17}$$

By existing analysis, we know that $\mathbb{E}[R^*(\frac{1}{n}\mathbf{Z}^T\boldsymbol{\epsilon})] \leq \frac{c_1}{\sqrt{n}}\omega(\Omega_R)$, along with suitable concentration around the expectation [14]. Therefore, the new component of the analysis focuses on the second term $R^*(\frac{1}{n}\mathbf{Z}^T\mathbf{W}\boldsymbol{\beta}^*)$, which is a consequence of the noise. For simplicity, we consider the case when $\mathbf{X}$ is an isotropic bounded sub-Gaussian vectors such that $\Sigma_{\mathbf{x}} = \mathbf{I}_{p\times p}$, with sub-Gaussian norm K_1, and $\mathbf{W}$ is composed of independent rows sampled from Subg$(0, \Sigma_{\mathbf{w}}, K_{\mathbf{w}})$. The following lemma provides a suitable upper bound for the expectation of the second term $R^*(\frac{1}{n}\mathbf{Z}^T\mathbf{W}\boldsymbol{\beta}^*)$. Note that lemma can be easily extended to anisotropic bounded sub-Gaussian $\mathbf{X}$.

Lemma 4.14 *Assume that the statistical parameter $\boldsymbol{\beta}^*$ has unit L_2 norm, and the matrices $\mathbf{X}$ and $\mathbf{W}$ consist of isotropic bounded sub-Gaussian entries with sub-Gaussian norm K_1. Then, the following upper bound holds for the expectation.*

$$\mathbb{E}_{\mathbf{X},\mathbf{W}}\left[R^*\left(\frac{1}{n}\mathbf{Z}^T\mathbf{W}\boldsymbol{\beta}^*\right)\right] \leq R(\boldsymbol{\beta}^*)\nu + K_1 c\frac{\omega(\Omega_R)}{\sqrt{n}} \tag{4.18}$$

$$+R(\boldsymbol{\beta}^*)\left[\frac{\eta_0\lambda_{\max}(\Sigma_{\mathbf{w}})\omega(\Omega_R)}{\sqrt{n}}\right] \tag{4.19}$$

where $\nu = \sup_{\mathbf{u}\in\Omega_R}\|\Sigma_{\mathbf{w}}^{1/2}\mathbf{u}\|_2^2$ and $c, c_2 > 0$ are constants.

Proof: Note that

$$\mathbb{E}\left[R^*\left(\frac{1}{n}\mathbf{Z}^T\;\mathbf{W}\boldsymbol{\beta}^*\right)\right] \leq \mathbb{E}\left[R^*\left(\frac{1}{n}\mathbf{X}^T\mathbf{W}\boldsymbol{\beta}^*\right)\right] + \mathbb{E}\left[R^*\left(\frac{1}{n}\mathbf{W}^T\mathbf{W}\boldsymbol{\beta}^*\right)\right] \; . \tag{4.20}$$

We upper bound the two terms as follows. First, consider the first term.

$$\mathbb{E}_{\mathbf{X},\mathbf{W}}\left[R^*\left(\frac{1}{n}\mathbf{X}^T\mathbf{W}\boldsymbol{\beta}^*\right)\right] = \mathbb{E}_{\mathbf{W}}\left[\frac{1}{n}\|\mathbf{W}\boldsymbol{\beta}^*\|_2\right]\mathbb{E}_{\mathbf{X}}\left[R^*\left(\mathbf{X}^T\mathbf{u}\right)\right] \tag{4.21}$$

where $\mathbf{u} = \mathbf{W}\boldsymbol{\beta}^*/\|\mathbf{W}\boldsymbol{\beta}^*\|_2 \in \mathbb{S}^{p-1}$ is an unit vector and since $\mathbf{X}$ and $\mathbf{W}$ are independent the expectation factorizes. Since $\mathbf{W}\beta^*$ and $\mathbf{X}^T\mathbf{u}$ are sub-Gaussian vectors with i.i.d. rows $(\mathbf{W}\boldsymbol{\beta}^*)_i$ and $(\mathbf{X}^T\mathbf{u})_i$, each of which is sub-Gaussian with sub-Gaussian norm smaller than K_1, we have:

$$\mathbb{E}_{\mathbf{W}}\left[\frac{1}{n}\|\mathbf{W}\boldsymbol{\beta}^*\|_2\right] \leq \frac{1}{n}K_1\sqrt{n} \tag{4.22}$$

$$\mathbb{E}_{\mathbf{X}}\left[R^*\left(\mathbf{X}^T\mathbf{u}\right)\right] \leq c\omega(\Omega_R) \; , \tag{4.23}$$

so that

$$\mathbb{E}_{\mathbf{X},\mathbf{W}}\left[R^*\left(\frac{1}{n}\mathbf{X}^T\mathbf{W}\boldsymbol{\beta}^*\right)\right] \leq K_1 c\frac{\omega(\Omega_R)}{\sqrt{n}} \tag{4.24}$$

Next, we consider the second term, and note that

$$\begin{aligned}
\mathbb{E}_{\mathbf{W}}\left[R^*\left(\frac{1}{n}\mathbf{W}^T\mathbf{W}\boldsymbol{\beta}^*\right)\right] &= \frac{1}{n}\mathbb{E}_{\mathbf{W}}\left[\sup_{\mathbf{u}\in\Omega_R}\langle \mathbf{W}\mathbf{u}, \mathbf{W}\boldsymbol{\beta}^*\rangle\right] && (4.25)\\
&\overset{(a)}{=}\frac{R(\boldsymbol{\beta}^*)}{n}\mathbb{E}_{\mathbf{W}}\left[\sup_{\mathbf{u}\in\Omega_R}\langle \mathbf{W}\mathbf{u}, \mathbf{W}\mathbf{v}\rangle\right] && (4.26)\\
&\overset{(b)}{\leq} R(\boldsymbol{\beta}^*)\mathbb{E}_{\mathbf{W}}\left[\sup_{\mathbf{u}\in\Omega_R}\frac{1}{n}\|\mathbf{W}\mathbf{u}\|_2^2\right] && (4.27)\\
& && (4.28)
\end{aligned}$$

where (a) follows from noting that $\mathbf{v} = \boldsymbol{\beta}^*/R(\boldsymbol{\beta}^*) \in \Omega_R$, and (b) follows from the inequality $2\langle \mathbf{W}\mathbf{u}, \mathbf{W}\mathbf{v}\rangle \leq \|\mathbf{W}\mathbf{u}\|_2^2 + \|\mathbf{W}\mathbf{v}\|_2^2$, and taking supremum over all $\mathbf{u} \in \Omega_R$.

[47] shows that if $\mathbf{W}$ consists of i.i.d. sub-Gaussian rows $\mathbf{w}_i \sim \text{Subg}(0, \Sigma_{\mathbf{w}}, K_{\mathbf{w}})$, then

$$\left|\frac{1}{n}\|\mathbf{W}\mathbf{u}\|_2^2 - \|\Sigma_{\mathbf{w}}^{1/2}\mathbf{u}\|_2^2\right| \leq \max(\delta, \delta^2) \quad \forall \mathbf{u} \in \Omega_R \tag{4.29}$$

with probability at least $1 - 2\exp(-\eta_1\tau^2)$, where $\delta = \frac{\eta_0\lambda_{\max}(\Sigma_{\mathbf{w}})\omega(\Omega_R)}{\sqrt{n}} + \frac{\tau}{\sqrt{n}}$, and η_0, η_1 are constants dependent on $K_{\mathbf{w}}$. Therefore, we obtain

$$\sup_{\mathbf{u}\in\Omega_R}\frac{1}{n}\|\mathbf{W}\mathbf{u}\|_2^2 \leq \nu + \frac{\eta_0\lambda_{\max}(\Sigma_{\mathbf{w}})\omega(\Omega_R)}{\sqrt{n}} + \frac{\tau}{\sqrt{n}}\ , \tag{4.30}$$

with probability at least $1 - 2\exp(-\eta_1\tau^2)$, where $\nu = \sup_{\mathbf{u}\in\Omega_R}\|\Sigma_{\mathbf{w}}^{1/2}\mathbf{u}\|_2^2$.

Therefore,

$$\mathbb{E}\left[R^*\left(\frac{1}{n}\mathbf{W}^T\mathbf{W}\boldsymbol{\beta}^*\right)\right] \leq R(\boldsymbol{\beta}^*)\left[\nu + \frac{\eta_0\lambda_{\max}(\Sigma_{\mathbf{w}})\omega(\Omega_R)}{\sqrt{n}}\right] \tag{4.31}$$

■

4.4.2 Proof of Lemma 4.9

Proof of this lemma follows the same line of proof of Theorem 4.5, except in this case instead of $R^*\left(\frac{1}{n}\mathbf{W}^T\mathbf{W}\boldsymbol{\beta}^*\right)$ we end up with $R^*\left(\frac{1}{n}\mathbf{W}^T\mathbf{W}\boldsymbol{\beta}^* - \frac{1}{n}\mathbf{W}_0^T\mathbf{W}_0\boldsymbol{\beta}^*\right)$ where $\mathbf{W}$ and $\mathbf{W}_0$ have same distributions and cancel out each others effects in expectation. Thus the statement follows.

4.4.3 Proof of Lemma 4.10

Proof: First we right the RE condition as follows:

$$\begin{aligned} & \inf_{\mathbf{v}\in\mathcal{A}_w} \mathbf{v}^T\hat{\Gamma}\mathbf{v} \\ = \; & \frac{1}{n}\mathbf{X}^T\mathbf{X} + \frac{1}{n}\mathbf{W}^T\mathbf{W} - \Sigma_{\mathbf{w}} + \Sigma_{\mathbf{w}} - \hat{\Sigma}_{\mathbf{w}} \\ = \; & \frac{1}{n}\mathbf{X}^T\mathbf{X} + \frac{1}{n}\mathbf{W}^T\mathbf{W} - \Sigma_{\mathbf{w}} + \Sigma_{\mathbf{w}} - \frac{1}{n}\mathbf{W}_0^T\mathbf{W}_0 \end{aligned} \tag{4.32}$$

Now we lower bound $\frac{1}{n}\mathbf{X}^T\mathbf{X}$, $\frac{1}{n}\mathbf{W}^T\mathbf{W} - \Sigma_{\mathbf{w}}$, and upper bound $\frac{1}{n}\mathbf{W}_0^T\mathbf{W}_0 - \Sigma_{\mathbf{w}}$. Note that rows of both $\mathbf{W}$ and $\mathbf{W}_0$ are iid sampled from same distribution. Therefore, we need lower and upper RE condition for $\frac{1}{n}\mathbf{W}^T\mathbf{W} - \Sigma_{\mathbf{w}}$. The result can be instantiated from Theorem 12 of [14] where we have following bounds with probability at least $(1 - 2\exp(-\eta_i\omega^2(\mathcal{A}_w)))$

$$\begin{aligned} \lambda_{\min}(\Sigma_{\mathbf{x}}|\mathcal{A}_w)\left(1 - c_1\frac{\omega(\mathcal{A}_w)}{\sqrt{n}}\right) & \leq \inf_{\mathbf{u}\in\mathcal{A}_w} \frac{1}{n}\|\mathbf{X}\mathbf{u}\|_2^2 \\ -c_2\lambda_{\min}(\Sigma_{\mathbf{x}}|\mathcal{A}_w)\frac{\omega(\mathcal{A}_w)}{\sqrt{n}} & \leq \inf_{\mathbf{u}\in\mathcal{A}_w} \frac{1}{n}\mathbf{W}^T\mathbf{W} - \Sigma_{\mathbf{w}} \\ c_2\lambda_{\max}(\Sigma_{\mathbf{x}}|\mathcal{A}_w)\frac{\omega(\mathcal{A}_w)}{\sqrt{n}} & \geq \sup_{\mathbf{u}\in\mathcal{A}_w} \frac{1}{n}\mathbf{W}^T\mathbf{W} - \Sigma_{\mathbf{w}} \end{aligned} \tag{4.33}$$

Putting together the inequities the lemma follows. ■

4.4.4 Proof of Theorem 4.12

Proof: We start from the optimality of $\hat{\boldsymbol{\beta}}_r$:

$$\begin{aligned} & \hat{\boldsymbol{\beta}}^T\hat{\Gamma}\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}^T\frac{1}{n}\mathbf{Z}^T\mathbf{y} + \lambda R(\hat{\boldsymbol{\beta}}) \\ & \qquad \leq \boldsymbol{\beta}^{*T}\hat{\Gamma}\boldsymbol{\beta}^* - \boldsymbol{\beta}^{*T}\frac{1}{n}\mathbf{Z}^T\mathbf{y} + \lambda R(\boldsymbol{\beta}^*) \\ \Rightarrow \; & \boldsymbol{\delta}^T\hat{\Gamma}\boldsymbol{\delta} \leq \boldsymbol{\delta}^T\left(\frac{1}{n}\mathbf{Z}^T\mathbf{y} - \hat{\Gamma}\boldsymbol{\beta}^*\right) + \lambda(R(\boldsymbol{\beta}^*) - R(\hat{\boldsymbol{\beta}})) \\ \Rightarrow \; & \boldsymbol{\delta}^T\hat{\Gamma}\boldsymbol{\delta} \leq \boldsymbol{\delta}^T\left(\frac{1}{n}\mathbf{Z}^T\mathbf{y} - \hat{\Gamma}\boldsymbol{\beta}^*\right) + \lambda R(\boldsymbol{\delta}) \end{aligned} \tag{4.34}$$

Equation (4.33) shows that the LHS is lower bounded, with probability at least $(1-2\exp(-\eta_*\omega^2(\mathcal{A}_w)))$ where $\eta_* > 0$ is a constant, by RE condition as $0 \leq \kappa\|\boldsymbol{\delta}\|_2^2 \leq \boldsymbol{\delta}^T\hat{\Gamma}\boldsymbol{\delta}$, where

$$\kappa = \lambda_{\min}(\Sigma_{\mathbf{x}}|\mathcal{A}_w)\left(1 - c_1\frac{\omega(\mathcal{A}_w)}{\sqrt{n}}\right) - c_2(\lambda_{\min}(\Sigma_{\mathbf{w}}|\mathcal{A}_w) + \lambda_{\max}(\Sigma_{\mathbf{w}}|\mathcal{A}_w))\frac{\omega(\mathcal{A}_w)}{\sqrt{n}}$$

is a positive constant when $n = O(\omega^2(\mathcal{A}_w))$. Next, we bound the first term of the RHS, $\frac{1}{n}\boldsymbol{\delta}^T\mathbf{Z}^T\mathbf{y}$ using Holder's inequality:

$$\begin{aligned} \boldsymbol{\delta}^T\left(\frac{1}{n}\mathbf{Z}^T\mathbf{y} - \hat{\Gamma}\boldsymbol{\beta}^*\right) &\leq R(\boldsymbol{\delta})R^*(\frac{1}{n}\mathbf{Z}^T\mathbf{y} - \hat{\Gamma}\boldsymbol{\beta}^*) \\ &\leq R(\boldsymbol{\delta})\lambda \end{aligned} \tag{4.35}$$

where the last inequality is from the definition of λ. Putting the bound back to the original inequality (4.34) we get:

$$\|\boldsymbol{\delta}\|_2^2 \leq 2R(\boldsymbol{\delta})\frac{\lambda}{\kappa} \leq 2\Psi(C_r)\|\boldsymbol{\delta}\|_2\frac{\lambda}{\kappa}, \tag{4.36}$$

and using Lemma 4.10 completes the proof. ∎

Chapter 5

Weighted Dictionary Learning for Twitter Sentiment Analysis

Social media have become an important part in the everyday life of millions of people around the world. Data is being produced by users with tremendous rate which is providing many scientific disciplines a wealth of data to analyze, with wide variety of applications. Recent studies [32, 31, 30, 84] empirically have shown the predictive value of social media content in domains such as marketing, business and politics.

Twitter data presents two fundamental and almost contradictory properties: it is at the same time overabundant at the global scale but scarce at the individual level. On the one hand, because of its widespread use and streaming nature, it can be considered as big data with all the computational constraints that this implies. On the other hand, the 140 characters limit (a maximum of about fifty words but only six on average, according to our experiments), severely restricts the amount of information per tweet, rendering the per-tweet analysis, the main goal of this paper, very challenging. The information is so scarce that sometimes even a human cannot analyze tweets accurately without prior information such as information about author. Including context knowledge, like geographic location and age may lead to different interpretation of a tweet.

Although twitter sentiment analysis increasingly gained attentions, there are several issues that limit its usage in practical applications. In general tweets do not always contain sentiments. They may contain information, facts or any other kind of objective expressions. Thus, before any sentiment analysis polar tweets (i.e., those with sentiment) should be separated from neutral ones. Knowing this fact, twitter analysis literature have moved from just sentiment analysis to considering neutral tweets in classification [85, 33].

Ignoring the objects, individuals or products that sentiment has been expressed about them is another major gap between current state of the art approaches and practical applications of twitter sentiment analysis. Because in practice we are interested in discovering people's feelings about a certain product, topic or in general a target [86]. There have been initial work on target-dependent sentiment analysis [86] which exploits history of users' tweets to do sentiment analysis.

Now we can delineate three steps required for sentiment analysis which are detecting tweets related to target of interest, separating tweets that have feelings and finally distinguishing sentiment types. The first step toward realistic sentiment analysis is topic classification, by which we mean distinguishing the tweets that are related to our topic of interest from unrelated ones [86]. This is a 2-class classification that has strong relation with topic modeling and classification. Topic modeling/classification are mature techniques when applied to usual texts [87, 88] and recent works have addressed topic modeling for Twitter data [89, 90, 91], but topic classification for micro-blogs has not been explored yet.

The second step is carried out with the goal of determining which tweets have emotional content (i.e., if it is subjective and express some kind of sentiment). This is sometimes referred to as the polar-neutral identification problem [33, 92, 93, 85].

Finally, sentiment analysis is performed only on tweets with emotional content. Many emotional dimensions can be extracted from rich text data such as weblogs, but because of the meager information contained in single tweets, they are usually classified into two main groups, according to their emotional energy: negative emotions (e.g., fear, hatred, resentment, anger, hostility), and positive emotions (e.g., enthusiasm, laughter, empathy, happiness).

Considering sentiment analysis as a three-step process of per-tweet classification is one approach. The other popular method which attempts to circumvent the scarcity of per-tweet information, is sentiment analysis for batches of tweets. These batches can be built using different criteria such as spatial (geographical location of senders), temporal (time of postings), or by history of author's posts. The common methods for batch analysis are lexicon-based, which use pre-compiled lists of polar words as indicators of the sentiment type [32, 31, 34, 35]. As expected, these approaches perform poorly on a per-tweet basis as later shown in our experiments. Overall, standard text analysis techniques are not suited to work with the limited data available in single tweets. Moreover, batch analysis methods are not suited for cases in which the grouping criteria are not trivial to obtain (e.g., when grouping by age, or gender), or are unknown (then the goal might actually be to find those groups). Finally, there is no criteria for validating the result of batch sentiment analysis methods in literature other than results that show obtained sentiments are aligned with world's events [32, 31].

This work is devoted to analyze the emotional content of *single* tweets and is, to the best of our knowledge, the first attempt to address all the aforementioned classification tasks together. In addition, as a matter of completeness, we supplement our work with results of aggregated tweets analysis. All tasks have been formulated as 2-class classification problems and several supervised learning methods have been used to perform classification. Unlike many previous works that uses sophisticated language features [92, 85, 33] with heavy pre-processing we just use bag-of-words as the input of classification. We also provide a new method for polar-neutral identification problem exploiting a fact from experimental psychology [34]. Also we experimentally show that classification can be performed on random reconstructible projection of high dimensional sparse input data without losing performance accuracy. Finally, we utilized the available soft labels, provided by aggregation of assigned labels by group of evaluators, and supplement our work with weighted variant of all presented classification methods.

The rest of this chapter is organized as follows. Labeling, pre-processing and classification algorithms are discussed in 5.1. In Section 5.2 we present experimental results and detailed comparisons of several methods.

5.1 The Classification Pipeline

In this section we discuss the whole classification process. We begin by explaining how the data is labeled for training. Then we comment on the parsing and preprocessing procedures which output tweets represented by high-dimensional feature vectors using a bag-of-words approach. Following literature [94, 95] and our preliminary experiment results, support of the bag-of-words vector is used as input feature vector. High dimension and sparsity of input vector enable us to use random reconstructible projection, to reduce its dimensionality. Experiments show that using the resulted vector is proper for classification purposes and also accelerates algorithms by reducing the computational complexity of operations.

5.1.1 Labeling the Data

Supervised learning algorithms obviously rely on the availability of labeled data. The use of specific words to label tweets is common. For example, [31] use the phrase “I feel” to label tweets as polar (i.e., expressing emotions) and [93] use an emoticon list as indicators for positive or negative content. Other works [96, 92] gather noisy labels from multiple sources, like unevaluated sentiment analysis tools, and then incorporate this uncertainty into the classification algorithm.

In the last few years, crowdsourcing has emerged as a cost-effective way to carry out labor-intensive tasks, thus becoming popular in the machine learning community [97]. In this work, the process of data labeling is crowdsourced as part of the Dialogue Earth Project,[1] and the data was kindly provided for the experiments of this report.

Since we are facing a dataset which is growing with the rate of 200 million data point per day it would be wise to prune it using simple techniques at the first step. The first task in the hierarchy of aimed tasks is topic classification. So it is reasonable to decrease data size based on goal of topic classification. Thus, we perform gross filtering on a collection of tweets based on an extensive list of words associated to the topic of interest and filter out tweets that do not contain any of indicator words. For example if the target topic is weather, by using a list of words that relates to weather (like snow, cold, hot etc.) we separate relevant tweets. But still there are many irrelevant tweets in our dataset which makes the topic classification task a necessity. We may also miss some topic-related tweets that do not contain any of our compiled words which is inevitable because of the size of data. In this article, we use databases collected for weather and gas price topics.

The data is then hand labeled by several trustworthy evaluators (i.e., people that consistently showed good accuracy during quality control tests) with 4 labels: positive, negative, neutral, and not related to the target topic. An additional label is reserved for cases in which the evaluator cannot assign a tweet to any of the aforementioned classes. It must be noted that the quality control tests ensure that the labels are not too noisy, and when an evaluator cannot label a tweet, it can be interpreted as if there is no context-independent information in it. It should be mentioned that having trustworthy evaluators makes our data source different from other

[1] www.dialogueearth.org

crowdsourced data whose annotation quality should be evaluated itself [97]. Thus, disagreement of evaluators shows the inherent difficulty of the task at hand.

Let $\mathcal{C}$ be the set of all classes. For each tweet i, evaluator j choose a label $\mathbf{L}_{ij}$ that is a $|\mathcal{C}|$ dimensional vector in which one element is equals 1 and all remaining elements are zero. By normalizing sum of these vectorial labels for each tweet i we get our soft vector label as $\boldsymbol{\omega}_i = (\sum_{j=1}^{m} \mathbf{L}_{ij})/m$ where $\boldsymbol{\omega}_i$ is the $|\mathcal{C}|$ dimensional label vector representing the confidence of each label for tweet i.

Having confidence vector $\boldsymbol{\omega}_i$ in hand we can work with two variants of label set. First one is just soft labels contained in each $\boldsymbol{\omega}_i$ which is $\mathcal{Y} = \{\omega_{ic} \in [0,1] \mid i = 1,\ldots,n; c = 1,\ldots,|\mathcal{C}|\}$, where ω_{ic} represents the confidence of label c for data point i. On the other hand we can consider hard labels derived from $\boldsymbol{\omega}_i$ that is $\tilde{\mathcal{Y}} = \{y_i = \mathrm{argmax}_{c\in\mathcal{C}}\omega_{ic} \mid i = 1,\ldots,n\}$, and thus falling back into usual classification configuration in which each data point has only single label. Since we have worked with both soft label $\mathcal{Y}$ and hard label $\tilde{\mathcal{Y}}$ we name the members of latter set dominant labels for convenience. Also when we discuss weighted algorithms we refer to algorithms that use soft label set $\mathcal{Y}$.

Finally, since in each three steps of sentiment analysis we perform 2-class classification on two subsets of $\mathcal{C}$, (e.g., related vs. not related which contains neutral, positive and negative) we should compute the weights of these subsets. Assume that we want to classify $\mathcal{C}_1, \mathcal{C}_2 \subset \mathcal{C}$ where $\mathcal{C}_1 \cap \mathcal{C}_2 = \emptyset$ and they are not necessarily partition $\mathcal{C}$. Then for each tweet i we should compute two weights $\omega_{i\mathcal{C}_1}$ and $\omega_{i\mathcal{C}_2}$ which are simply sum of the labels' weights that are present in $\mathcal{C}_1$ and $\mathcal{C}_2$: $\omega_{i\mathcal{C}_j} = \sum_{c_k\in\mathcal{C}_j} \omega_{ik}, j = 1,2$.

5.1.2 Parsing and Preprocessing

In this work, we use the bag-of-words model for representing the tweets. One of the main advantages of this approach is that the parsing procedure is very simple. Many previous works have included language level features, like part of speech tags in input [93, 92, 33, 85] which makes the pre-processing step complicated.

We begin by extracting the words in a tweet. Here we use the term word in a broad sense, which for us encompasses actual words but also numbers, usernames, emoticons, URLs, etc. Some of them however receive special treatment. We do not care about the actual value/content of numbers, usernames, and URLs, and thus replace them by special generic identifiers. Notice that simply removing these words would be harmful, since these type of words are common in neutral tweets which just share information between the sender and the receivers. We do remove re-tweet signs (RT), special characters (not contained in emoticons), and stop words (e.g., ‘the,’ ‘of,’ ‘about’). Note that we have removed all polar words (e.g., ‘great,’ ‘bad,’ ‘better’) from stop word list to prevent loss of emotional signals. Also all negation words (e.g., ‘not,’ ‘never’) have been removed from stop word list, because they can change the sentiment completely [93]. Hashtags are a special kind of keywords used in Twitter that are often a concatenation of words (e.g., ‘#rainymorning,’ ‘#loveThisWeather’); when the words in a hashtag begin with an uppercase character we break it into separate words.

We automatically spell check the words using three dictionaries: an English dictionary, a Twitter dictionary which contains specific lingo, and an emoticon dictionary. The Twitter dictionary has been gathered from several online Twitter dictionaries that list popular words coined by Twitter users. After spell checking we remove words that are not in any of the mentioned dictionaries. We found that stemming did not improve the classification results and thus we omit it from the preprocessing procedure.

After cleaning raw tweets using all of above steps, we perform another step which empirically proved to be effective in both speed and accuracy of classifications. Words that appeared in cleaned database less than thrice are pruned. Also based on the desired task, highly frequent non-distinctive words are being removed. A word named high frequent in a class, if it appears on average more than a “high frequency threshold” in each tweet of that class. Among high frequent words of two classes those with average frequency closer than “similarity threshold” are called non-distinctive.

5.1.3 Representing and Compressing Tweets

The bag-of-words model is one of the most commonly employed feature extraction approaches in text (and image) classification. A text (document) is simply represented as an unordered collection (i.e., a set that may contain repeated elements) of words $\mathcal{W}$. A predefined set of words $\mathcal{L} = \{l_i \mid i = 1, \ldots, d\}$ is then used to build an d-dimensional feature vector $\mathbf{v}$ for each document W such that $(\forall i = 1, \ldots, d)$ $\mathbf{v}(i) = \#(\mathcal{W}, l_i)$, where $\#(\mathcal{W}, l_i)$ is the number of times word l_i appears in $\mathcal{W}$.

Usually d is counted in the tens of thousands (e.g., in our experiments $d \approx 10^4$), but since the length of a tweet is limited to 140 characters, the bag-of-words approach produces extremely sparse feature vectors when dealing with twitter data.

Although we have done experiments using original bag-of-words data, we extend our experiments and also use low dimensional projection of it as input. Working in original high-dimensional domain impose computational difficulties which naturally lead to dimensionality reduction techniques. Here we show that working with (extremely) sparse d-dimensional feature vectors directly is not necessary, one can instead reduces their dimensionality using random reconstructible projection and perform classification in the resulted domain without considerable loss of accuracy. Since random reconstructible projection is a well-know technique in compressed sensing literature [40] we use compression and projection terms interchangeably. Recent result [98] shows theoretically that learning can be done in compressed domain without significant loss in classification accuracy for support vector machine. In this paper we show empirically that compressed learning (i.e., learning in compressed domain) is also possible for other well-known classification algorithms.

In this framework an $m \times d$ matrix $\mathbf{P}$ ($m \ll d$) is used to create a compressed representation $\mathbf{x} = \mathbf{P}\mathbf{v}$ of a feature vector $\mathbf{v}$ in such a way that m is as small as possible and $\mathbf{v}$ can be reconstructed from $\mathbf{x}$. The best reconstruction performance is obtained when $\mathbf{P}$ is a random matrix, i.e., when its entries p_{ij} are sampled from i.i.d. random variables [99]. In this paper we build $\mathbf{P}$ by sampling its entries p_{ij} from a Gaussian distribution $\mathcal{N}(0, 1/m)$. The value of m is chosen such that $m = O(h \log(d/h))$ where $h = \max_{\mathbf{v} \in V} \|\mathbf{v}\|_0$ (while $d \approx 10^4$, $h \approx 20$). We have tried other types of methods for generating $\mathbf{P}$ [99] in preliminary experiments and based on performance chose to work with Gaussian random projection.

To conclude, the set of (one per-tweet) feature vectors $\mathcal{V} = \{\mathbf{v}_i \mid i = 1, \ldots, n\}$ is represented in the compressed domain by a set of vectors $\mathcal{X} = \{\mathbf{x}_i \mid i = 1, \ldots, n\}$, where $\mathbf{x}_i = \mathbf{P}_{m \times d}\mathbf{v}_i$. The goal is now to learn to classify the tweets based on this compressed representation.

It worth mentioning that random projection (RP) has been used as a dimensionality reduction technique previously in the literature [100] and its classification performance has been compared with other dimensionality reduction methods like PCA [101]. The result of this comparison is that PCA outperforms RP but RP is computationally more efficient. In this work, we also used PCA and surprisingly it underperforms random reconstructible projection (RRP) in almost all classification methods. This is in accordance with the theoretical result of [98] that shows when original input vector is sparse classification with high accuracy in compressed domain constructed by RRP is possible.

5.1.4 Classification Methods

Several well-known supervised learning algorithms and their weighted variant have been chosen for performing classification. Among them Support Vector Machine(SVM) [102], K Nearest Neighbor(KNN) [25] and Naïve Bayes(NB) [25] are well known. Hence, we only explain the classification algorithm which is based on dictionary learning [58] in detail, along with brief descriptions of weighted variants of other methods.

Sparse Modeling Approach to Classification

For each step of tweet classification $|\mathcal{C}| = 2$ and $\mathcal{Y}$ is the set of binary values that represent dominant label of each data point. For utilizing the possible available information in the non-binary confidence ω_{ic} introduced in 5.1.1, we propose to redefine the cost function for each datum $\mathbf{x}_i$ and each class dictionary $\mathbf{D}_c$ as

$$\ell_\omega(\mathbf{x_i}, \mathbf{D}_c) = \min_{\boldsymbol{\alpha}} \ \omega_{ic} \left[\frac{1}{2} \|\mathbf{x} - \mathbf{D}_c \boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1 \right].$$

thereby using this cost instead of ℓ when learning each class dictionary. The contribution of the sample $\mathbf{x}_i$ to the class c is then weighted by its non-binary label $\omega_c(\mathbf{x}) = \omega_{ic}$. The closer ω_{ic} is to one, the more $\mathbf{x}_i$ contributes to class c. In the extreme case that $\omega_{ic} = 0$, $\mathbf{x}_i$ does not contribute at all to the learning of the dictionary for class c.

We then solve the optimization problem

$$\min_{\mathbf{D}} \ \frac{1}{n} \sum_{i=1}^{n} \ell_\omega(\mathbf{x}_i, \mathbf{D}) \tag{5.1}$$

for each class c, by alternating the minimization over $\mathbf{D}$ and the sparse codes $\boldsymbol{\alpha}_i$.

Sparse coding: Minimizing Equation (5.1) over the sparse codes $\boldsymbol{\alpha}_i$ with $\mathbf{D}$ fixed involves solving for each $i = 1, \ldots, n$,

$$\min_{\boldsymbol{\alpha}_i} \ \omega_{ic} \left[\frac{1}{2} \|\mathbf{x}_i - \mathbf{D} \boldsymbol{\alpha}_i\|_2^2 + \lambda \|\boldsymbol{\alpha}_i\|_1 \right].$$

Since for each subproblem ω_{ic} is constant, this is a classical LASSO problem and there is no need for designing particular minimization techniques and methods like LARS [103] can be used directly.

Dictionary learning: Because of its streaming nature and widespread use, twitter data can be considered as a massive data source. Therefore, online learning algorithms arise as an obvious choice for analyzing them. Following the online dictionary learning approach of [23], for minimizing Equation (5.1) over $\mathbf{D}$ (one such $\mathbf{D}_c$ per class c), with the sparse codes $\boldsymbol{\alpha}_i$ fixed for all i, we rewrite it as

$$\min_{\mathbf{D}} \frac{1}{n}\left(\frac{1}{2}\mathrm{Tr}(\mathbf{D}^T\mathbf{D}\mathbf{A}) - \mathrm{Tr}(\mathbf{D}^T\mathbf{B})\right) \tag{5.2}$$

where $\mathbf{A} = \sum_{i=1}^{n} \omega_i \ \boldsymbol{\alpha}_i\boldsymbol{\alpha}_i^T$ and $\mathbf{B} = \sum_{i=1}^{n} \omega_i \ \mathbf{x}_i\boldsymbol{\alpha}_i^T$ (since in this case for simplicity we dropped the subindex c from $\mathbf{D}$, for consistency we write ω_i when meaning ω_{ic}). [23] have shown that there is a closed form for updating each column of $\mathbf{D}$, and this also follows when we add the weights as in Equation (5.2).

The complete optimization scheme for this online weighted dictionary learning algorithm is depicted in Algorithm 2 (recall that we learn one dictionary per class). The implementation is obtained by adding the weights to the publicly available SPAMS library.[2]

We wrap up dictionary learning classification with a discussion about dictionary learning in original (uncompressed) domain. Dictionary learning in original domain would try to minimize $\ell(\mathbf{v}_i, \mathbf{D}) = \frac{1}{2}\|\mathbf{v}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda\|\boldsymbol{\alpha}_i\|_1$ for each feature vector $\mathbf{v}_i$. But it is well known that for l_2 penalty we assume that error $\mathbf{v}_i - \mathbf{D}\boldsymbol{\alpha}_i$ is Gaussian noise [104] and since text documents does not usually satisfy this assumption, in this article we only perform dictionary learning in compressed domain. Recently Kasiviswanathan et al. [91] have presented an alternative formulation using l_1 reconstruction error for novel topic detection in twitter.

Naïve Bayes

Naïve Bayes classifier in its general form assign to each test data point the maximum a posteriori class

$$y_i = \operatorname*{argmax}_{c\in C} P(c|\mathbf{v}_i).$$

Using Bayes' rule $P(c|\mathbf{v}_i) = P(c)P(\mathbf{v}_i|c)/P(\mathbf{v}_i)$ and the fact that $P(\mathbf{v}_i)$ is constant for all classes we will have

$$y_i = \operatorname*{argmax}_{c\in C} P(c)P(\mathbf{v}_i|c)$$

[2] http://www.di.ens.fr/willow/SPAMS/

Algorithm 2 Weighted Online Dictionary Learning [23]

```
input: a random variable $\mathbf{x} \in \mathbb{R}^m$ with p.d.f. $p(\mathbf{x})$ (the training data), a weighting function $\omega : \mathbb{R}^m \to [0, 1]$, a regularization parameter $\lambda \in \mathbb{R}$, an initial dictionary $\mathbf{D}_0 \in \mathbb{R}^{m \times k}$, the number of iterations $T$.
output: the dictionary $D_T$.
$\mathbf{A}_0 \in \mathbb{R}^{k \times k} \leftarrow 0$ , $\mathbf{B}_0 \in \mathbb{R}^{m \times k} \leftarrow 0$
for $t = 1$ to $T$ do
    Draw $\mathbf{x}_t$ from $p(\mathbf{x})$
    Sparse coding: compute (e.g., using LARS)
        $$\alpha_t = \underset{\alpha \in \mathbb{R}^k}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{x}_t - \mathbf{D}_{t-1}\boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1.$$
    $\mathbf{A}_t \leftarrow \mathbf{A}_{t-1} + \omega(\mathbf{x}_t)\ \boldsymbol{\alpha}_t \boldsymbol{\alpha}_t^T$
    $\mathbf{B}_t \leftarrow \mathbf{B}_{t-1} + \omega(\mathbf{x}_t)\ \mathbf{x}_t \boldsymbol{\alpha}_t^T$
    Compute $\mathbf{D}_t$ with $\mathbf{D}_{t-1}$ as warm restart, solving
        $$\begin{aligned} \mathbf{D}_t &= \underset{\mathbf{D}}{\operatorname{argmin}} \frac{1}{t} \sum_{i=1}^{t} \omega(\mathbf{x}_t) \left[ \frac{1}{2} \|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda \|\boldsymbol{\alpha}_i\|_1 \right] \\ &= \underset{\mathbf{D}}{\operatorname{argmin}} \frac{1}{t} \left( \frac{1}{2} \mathrm{Tr}(\mathbf{D}^T \mathbf{D} \mathbf{A}_t) - \mathrm{Tr}(\mathbf{D}^T \mathbf{B}_t) \right). \qquad (5.3) \end{aligned}$$
end for
```

which will be simplified by assuming the conditional independence of input vector's features:

$$y_i = \operatorname*{argmax}_{c \in C} P(c) \prod_{j=1}^{d} P(v_{ij}|c), \tag{5.4}$$

where $P(c)$ and $P(v_{ij}|c)$s are computed from their corresponding frequencies in training data.

Now we should incorporate confidence weights of data points in Naïve Bayes formulation. So instead of computing $P(c)$ using class frequency we use weighted class frequency

$$P(c) = \frac{\sum_i \omega_{ic}}{\sum_c \sum_i \omega_{ic}},$$

in which each data point $\mathbf{v}_i$ contributes to the class c's probability the amount that is proportional to its label confidence ω_{ic}. Also for each feature $P(v_{ij}|c)$ probability should be computed based on weights of data:

$$P(v_{ij}|c) = \frac{\sum_i \omega_{ic} \times v_j}{\sum_i \omega_{ic}}.$$

K Nearest Neighbor

In K Nearest Neighbor class label is assigned to each test data point based on the labels of K closest training examples in the feature space. In order to use weight information in KNN instead of majority voting between K nearest neighbor of $\mathbf{v_i}$ we add their K confidence vector and pick the label with highest confidence:

$$y_i = \operatorname*{argmax}_{c \in C} \sum_{j \in \text{KNN}(i)} \omega_{jc}$$

Support Vector Machine

Support vector machine (SVM) tries to find a separating hyperplane which maximizes the margin between two classes. In its original formulation following optimization should be solved

$$\mathbf{W}^* = \operatorname*{argmin}_{\mathbf{W}} \Phi(\mathbf{W}) = \frac{1}{2}||\mathbf{W}||^2 + B \sum_{i=1}^{n} \xi_i \quad \text{s.t.}$$

$$y_i(\langle \mathbf{W}, \phi(\mathbf{v}_i) \rangle) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, ..., n,$$

where W is the normal vector to the hyperplane and ϕ maps v_i to higher dimensional space. Constant B determines the trade off between margin maximization and classification violation. For minimizing $\Phi(W)$ one can maximize its dual

$$\boldsymbol{\alpha}^* = \underset{\boldsymbol{\alpha}}{\operatorname{argmax}} W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

$$\text{s.t.} \quad \sum_{i=1}^{n} y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq B, \quad i = 1, ...n,$$

where $K(x_i, x_j) = (\langle \phi(x_i)\phi(x_j) \rangle)$ is the kernel.

For including weight in SVM we are following [105] which introduced weighted SVM (WSVM) to decrease the effect of outliers in SVM. In [105] they have weights from kernel-based possible c-means, but here we use the confidence scores of two classes (C_1 and C_2) that we want to do classification for them as weights.

The key idea is that for point v_i that we are sure about its label (i.e., $|w_{iC_1} - w_{iC_2}|$ is near 1) we should have larger penalty which reinforce correct classification more that margin maximization. So primal will change to the following

$$\mathbf{W}^* = \underset{\mathbf{W}}{\operatorname{argmin}} \Phi(\mathbf{W}) = \frac{1}{2}||\mathbf{W}||^2 + B\sum_{i=1}^{n} |\omega_{iC_1} - \omega_{iC_2}|\xi_i$$

$$\text{s.t. } y_i(\langle \mathbf{W}, \phi(\mathbf{v}_i) \rangle) + b) \geq 1 - \xi_i, \ \xi_i \geq 0, \quad i = 1, ..., n.$$

And relatively the dual will change to:

$$\boldsymbol{\alpha}^* = \underset{\boldsymbol{\alpha}}{\operatorname{argmax}} W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

$$\text{s.t.} \sum_{i=1}^{n} y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq |\omega_{iC_1} - \omega_{iC_2}|B, \quad i = 1, ...n.$$

As it is clear the only difference of WSVM with SVM is in the upper bound of box constraint for each lagrange multiplier.

Testing Procedure

Once labels have been assigned to the testing data following the classification procedure, a loss function is usually used to determine the accuracy of the assignment. When the label set is binary (i.e., each datum belongs to only one class), testing data $X_{\text{test}} = \{\mathbf{x}_i \mid i = 1, \ldots, n_{\text{test}}\}$ is accompanied by a label set $\tilde{Y}_{\text{test}} = \{y_i = \text{argmax}_{c \in C} \omega_{ic} \mid i = 1, \ldots, n_{\text{test}}\}$, and the per-sample loss function is usually

$$\mathbb{1}_{[f(\mathbf{x}_i) \neq y_i]},$$

where f is the mapping of input to output produced by classification algorithm and $\mathbb{1}_{[\bullet]}$ is the indicator function. Therefore, classification error is defined as

$$\frac{\sum_{i=1}^{n} \mathbb{1}_{[f(\mathbf{x}_i) \neq y_i]}}{n}.$$

In our weighted framework, the testing label set takes the form $Y_{\text{test}} = \{\omega_{ic} \mid i = 1, \ldots, n_{\text{test}}; c = 1, \ldots, C\}$. As mentioned in Section 5.1.1, at each step needed for sentiment analysis we perform 2-class classification to classify two disjoint subsets of C namely C_1 and C_2. So the weighted per-tweet loss would be

$$\omega_{iC_d} \cdot \mathbb{1}_{[f(\mathbf{x}_i) \neq C_d]}, \tag{5.5}$$

instead of regular loss where d is computed as

$$\omega_{iC_j} = \sum_{c \in C_j} \omega_{ic},\ d = \underset{j}{\text{argmax}}\, \omega_{iC_j},\ j = 1, 2.$$

Here we reiterated the definition of ω_{iC_j} from Section 5.1.1. In words, based on the task we aggregate the weights to only two weights ω_{iC_1} and ω_{iC_2}. Then we consider the bigger one as the label of the data point. The higher the weight of a datum is, the more it costs to classify it mistakenly. Accordingly, we should redefine the error as the total loss over all data, normalized by total possible loss:

$$\frac{\sum_{i=1}^{n} \omega_{ic_d} \cdot \mathbb{1}_{[f(\mathbf{x}_i) \neq c_d]}}{\sum_{i=1}^{n} \omega_{ic_d}}.$$

Note that prior for weighted methods should also be computed accordingly.

5.2 Experimental Validation

For the main part of the experiments, we used three collections of tweets. Two of them (DB1, DB2) are about weather and the last one is about gas price (GP) and they encompass 4490, 8850 and 12770 tweets respectively. We have used DB2 to adjust and validate our parameters. Then based on the best setting of parameters we perform experiments for all databases.

The databases are built by first collecting tweets with the Twitter API.[3] After crude filtering based on topic related word list, a few human evaluators, as explained in Section 5.1.1, were asked to assign to each tweet one of the following $C = 5$ classes:

- "Not related:" the tweet is not about the target topic;
- "Neutral:" the tweet contains no emotion;
- "Positive:" the tweet reflects positive feelings;
- "Negative:" the tweet reflects negative feelings;
- "I can't tell:" none of the above can be assessed.

Finally each tweet i receives a soft label, a weight ω_{ic} for each class $c = 1 \ldots 5$, equal to the proportion of evaluators that have chosen that class for the tweet. As commented in Section 5.1.1, notice that the "I can't tell" class simply means that the evaluator could not assign the tweet to any of the other classes, thus indicating the tweet actually has no label. We therefore discard from our analysis those tweets for which the "I can't tell" class gets the maximum weight, and we have $C = 4$ classes.

Next, preprocessing steps are performed as explained in Section 5.1.2. We picked 0.05 as high frequency threshold and select words that appeared on average more that 0.05 in each tweet as the candidate for removal. On average less than 30 words satisfy this condition in all databases. Then we remove those candidates that are close to each other. Here we consider words close if their distance in average frequency sense (i.e., similarity threshold defined in Section 5.1.2) is less than 0.2. We found that although numbers and links are high frequent in many tasks, they are usually distinctive (i.e., not close) for tasks involving neutral tweets. One possible explanation is that the neutral tweets which are about weather and gas price usually share information that contains numbers and a link to the source of information.

After preliminary phases, we will have 4-class classification problem. One approach is to perform multi-class classification. But since these classes have a natural hierarchical structure, we can solve the classification problem with a cascade type of approach. The steps in the cascade are the following: (1) topic classification, filtering out "not related" tweets; (2) polar-neutral classification; and (3) sentiment classification. We will present results for all these steps.

[3] https://dev.twitter.com/

We present the results obtained with Dictionary Learning (DL), Support Vector Machines (SVM), Nearest Neighbors (KNN), Naïve Bayes (NB) and weighted variants of all of them. For DL and WDL, the number of atoms and λ were fixed for all experiments.

Although in natural scenario, first task is topic classifications, because sentiment analysis is at the center of attention in literature, we start from it and subsequently discuss other tasks. Also we use sentiment analysis as a platform to explain our further parameters and their assigned values. All reported results are obtained using 10-fold cross validation.

5.2.1 Classifying Tweets by Sentiment

In this section we focus on sentiment analysis. In order to test the performance of the algorithms for this single task, we only consider tweets which have an associated positive or negative sentiment. For unweighted experiments we consider only dominant labels and for weighted experiments we use aggregated weights ω_{iC_1} and ω_{iC_2}, all defined in Section 5.1.1.

For each algorithm different parameter settings have been verified and results of best configurations are reported. Multinomial Naïve Bayes (MNB) outperformed other variants of NB in original domain, and in compressed domain using kernel density estimation was most helpful. K for KNN is set to 10 and since l_1 and l_2 distance metrics' performances were similar we report only results of l_2 distance. Linear kernel SVM performed better than other kernels like RBF, quadratic and polynomial. Also results of SVM in original and compressed domains were very close to each other which is what we expected from [98] theoretical guarantee. The main parameter in DL and WDL is the number of atoms in the dictionary. Our experiments showed that under-complete dictionaries (i.e., tall matrices) yield to higher accuracy. We introduced ratio parameter for dictionary to control the ratio of number of atoms to length of atoms (i.e, feature vector length). For all experiments we set aspect ratio to 0.5. All weighted experiments are done with the same parameter setting of their unweighted variants.

We consider two ways that input vector can be modified for experiments. First we can use support (i.e., binary version) of original word-count vector. Notice that tweets are themselves near binary, but empirically using support of input vectors improves the classification accuracy for all methods. Secondly, each of word-count or binary vectors can be projected to a compressed domain using random projection. So we end up with four different configurations for input vector. The result of each setting is presented for sentiment analysis task in Table 5.1 just for DB2. Based on theoretical reasons explained in 5.1.4 we do not perform DL in original

Table 5.1: Classification results for the positive vs. negative experiment for DB2 with different version of input vector. Prior of the experiment is 64.44%.

Binary?	Compressed?	SVM	NB	KNN	DL
True	True	79.19	75.04	74.50	78.94
True	False	80.16	82.95	75.01	-
False	True	77.67	71.49	73.60	77.02
False	False	76.86	81.33	74.17	-

domain. In addition we compared RRP with PCA in our preliminary experiments with similar destination dimension. In all classification methods except SVM, RRP outperform PCA. Since computational cost of PCA for large feature vector is high and increase in SVM's performance using PCA in comparison with RRP is negligible (less than 2%) we report only results of RRP dimensionality reduction.

Based on result of this step we picked for each method the setting for input vector which yields to best accuracy. NB and KNN best results are achieved with uncompressed binary vectors. SVM performance and speed increased using compressed binary vectors. Finally DL perform better when fed with binary vectors. From here on we only report results with these settings. Table 5.2 shows these results for all three databases. Note that prior of GP is much higher than other databases this is the reason why all methods perform better for GP database. It is interesting that most of tweets pertaining to gas price are negative.

We also compare our results with a lexicon-based methods specifically designed for twitter sentiment analysis . Following an extensive (and crowdsourced) study of words' sentiment in [35], they generated a list of words with happiness score from 1 to 10. After eliminating neutral words (i.e., with score around 5), they compute the weighted average happiness (WAH) of a batch of tweets by also taking into account word frequencies. The presented method [35] has been implemented and tested for our databases. As expected WAH is not proper for per-tweet tasks based on Table 5.2. As shown in Table 5.2, NB almost always outperforms other methods and KNN always has the worst performance.

Table 5.3 presents the results of weighted variants of different algorithms for weighted loss functions introduced in 5.1.4. It worth mentioning that when we train weighted algorithms with weighted data but use binary loss, accuracy stays the same or slightly improve from the case

Table 5.2: Classification results of unweighted algorithms for the positive vs. negative experiment for all databases with binary loss.

	DB1	DB2	GP
DL	78.72 ± 2.52	78.94 ± 0.96	86.46 ± 1.15
SVM	78.99 ± 2.65	79.19 ± 1.79	**87.34 ± 1.46**
KNN	75.20 ± 2.97	75.01 ± 2.30	86.88 ± 1.28
NB	**82.23 ± 3.24**	**82.95 ± 2.10**	87.29 ± 1.25
WAH	59.55	75.01	19.43
Prior	51.72%	64.44%	83.29%

Table 5.3: Classification results of weighted algorithms for the positive vs. negative experiment for all databases with weighted loss.

	DB1	DB2	GP
WDL	**81.12 ± 2.97**	81.43 ± 1.82	86.50 ± 1.02
WSVM	78.84 ± 3.77	82.13 ± 1.58	87.53 ± 1.18
WKNN	76.34 ± 3.80	78.92 ± 1.76	86.32 ± 1.62
WNB	80.35 ± 2.93	**83.28 ± 2.34**	**88.01 ± 1.28**
Prior	73.40%	56.99%	83.28%

in which unweighted algorithm are used with binary loss. Since the improvements are less than 2% we omit them from tables and only present weighted loss. Note that weighted priors of Table 5.3 is different from unweighted prior of Table 5.2. Here again WNB has the highest accuracy in almost all databases but its margin with WSVM and WDL is reduced in comparison with the accuracy margin of NB.

Also in this step we investigated the effect of projections. In order to mitigate possible “randomness”-related effects, we used ten different random projection matrices $\mathbf{P}$ and then merge the results (by using majority voting). Since the accuracy of each method appeared robust to number of projection, less than 1% variation was observed, we use only one projection from here on.

Table 5.4: Classification results of unweighted algorithms for the polar vs. neutral experiment for all databases with binary loss.

	DB1	DB2	GP
DL	80.29 ± 2.56	82.19 ± 1.65	74.00 ± 1.25
SVM	77.53 ± 2.35	79.80 ± 1.39	73.94 ± 1.50
KNN	74.26 ± 1.94	78.49 ± 1.88	70.47 ± 1.38
NB	**80.77 ± 2.00**	**82.53 ± 1.49**	**74.77 ± 1.15**
Prior	59.95%	58.22%	50.06%

Table 5.5: Classification results of weighted algorithms for the polar vs. neutral experiment for all databases with weighted loss.

	DB1	DB2	GP
WDL	84.29 ± 2.66	85.50 ± 1.31	74.37 ± 1.38
WSVM	81.92 ± 2.86	84.45 ± 1.20	73.43 ± 0.95
WKNN	80.49 ± 2.78	82.89 ± 1.14	**70.44 ± 1.46**
WNB	**84.58 ± 2.04**	**86.04 ± 1.46**	74.14 ± 1.59
Prior	59.10%	61.96%	50.06%

5.2.2 Detecting tweets with sentiment contents and topic classification

Different algorithms have been recently applied to sentiment analysis and polar-neutral classification, such as NB [93], SVM [92] and AdaBoost.MH [85]. All these approaches use rich feature vectors, that incorporate higher-level grammatical or semantical knowledge of some form. However we show that high accuracy can be achieved even with simple bag-of-words approach.

As in the previous section, we assume to have an oracle that discards tweets not related to the topic of interest. We therefore use only tweets for which the dominant label is positive, negative or neutral. Results for unweighted algorithms are shown in Table 5.4 and weighted algorithms' results with none-binary loss function are presented in Table 5.5. In both cases NB (WNB) performance is the best and is followed closely by DL (WDL).

Table 5.6: Classification results for the polar vs. neutral experiment for all databases using positive vs. background and negative vs. background classifiers.

	DB1	DB2	GP
DL	**79.95**	79.51	72.00
SVM	71.41	74.34	**73.67**
KNN	68.59	71.97	68.42
NB	78.56	**81.56**	72.78
Prior	59.95%	58.22%	50.06%

Table 5.7: Classification results of unweighted algorithms for topic classification experiment of weather databases with binary loss.

	DB1	DB2
DL	80.85 ± 2.12	81.15 ± 1.15
SVM	80.00 ± 1.04	78.73 ± 1.58
KNN	77.04 ± 2.03	75.51 ± 2.51
NB	**82.64 ± 1.93**	**81.93 ± 1.43**
Prior	72.24%	72.06%

Experimental psychology studies show that positive and negative sentiments are not opposite extremes of the same dimension but are, on the contrary, independent [34]. Based on this point, we introduce a new method for polar-neutral classification. We train two independent classifiers for separating positive (negative) tweets from all the rest. Then we use a simple aggregation scheme for classifying neutral and polar tweets. If a tweets is classified as positive or negative by one of the mentioned classifier it will be polar otherwise neutral. Results this classification scheme are shown in Table 5.6. Although the resulted accuracies are not as high as direct polar-neutral classification, quality of results shows possible promising direction.

We now turn our attention to the detection of tweets belonging to a given topic of interest. Since required label for GP database is not available we only report results of DB1 and DB2. Tables 5.7 and 5.8 show the results of topic classification for unweighted and weighted algorithms respectively. Again NB (WNB) and DL (WDL) are competing for the best performance.

Table 5.8: Classification results of unweighted algorithms for topic classification experiment of weather databases with binary loss.

	DB1	DB2
WDL	83.07 ± 2.29	**82.85 ± 1.14**
WSVM	82.55 ± 2.04	81.18 ± 1.69
WKNN	79.28 ± 2.39	73.32 ± 1.50
WNB	**83.93 ± 2.08**	82.59 ± 1.03
Prior	74.75%	74.85%

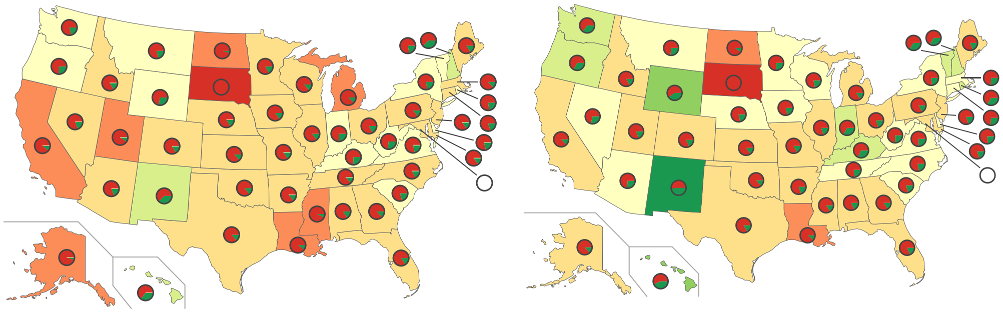
Table 5.9: Statistics of the error per state for the agglomerated WDL result. SD stands for standard deviation.

	Error (in %)			
	mean	SD	min	max
GP	5.21	3.71	0.00	13.76

5.2.3 Spatially Aggregated Results

We close the experimental section by showing spatially agglomerated results of the positive/negative classification. As mentioned at the begining of the chapter, sentiment analysis sometimes is being done on batches of tweets instead of single tweet. One common way of making batches is aggregation of tweets based on the geographic location of the authors (e.g., state or county). In this way we can discover the sentiment of people in that location, maybe in specific time, and interpret it.

Despite the fact that our methods are not specifically designed for processing batches of tweets, batch results are easily obtained once the individual classification is done. For this experiment, we only use the database for which the topic is gas prices. Figures 5.1 shows, the results aggregated per state for the gas price database using WDL and the ground truth map. Both maps and the additional statistics provided in Table 5.9 show that the state mood is correctly recovered by the proposed classification procedure.



(a) Ground truth map.

(b) Aggregated WDL result.

Figure 5.1: Comparison of WDL aggregated result with ground truth map. Red and green represent negative and positive sentiments respectively.

Part III

Discrete Problems

Chapter 6

Influence Maximization in Non-progressive Models

Motivated by viral marketing and other applications, the problem of influence maximization in a social network has attracted much attention in recent years. Given a social network where nodes represent users in a social group, and edges represent relationships and interactions between the users (and through which they influence each other), the basic idea of influence maximization is to select an initial set of "most influential" users (often referred to as the *seeds*) among all users so as to maximize the total influence under a given diffusion process (often referred to as the *influence model*) on the social network. In the context of viral marketing, this amounts to by initially targeting a set of influential customers, e.g., by providing them with free product samples, with the goal to trigger a cascade of influence through "word-of-mouth" or recommendations to friends to maximize the total number of customers adopting the said product. Domingos and Richardson [106] introduced this algorithmic problem to the Computer Science community and Kempe et al. [24] made the topic vastly popular under the name of *influence maximization*. They studied two influence models, the independent cascade (IC) model and the linear threshold (LT) model, and applied a greedy method to tackle the influence maximization problem [24]. Unfortunately Kempe et al.'s approach [24] for calculating the influence spread is based on Monte Carlo simulations which does not scale to large networks [60, 61]. As the result, it motivated researchers to either improve the scalability [60, 61] or study more tractable influence models [63, 64].

The focus of almost all of these earlier studies are, however, *progressive* influence models, including LT and IC models, in which once a costumer adopts a product or a user performs an action she cannot revert it. Retweeting news and visiting a commercial webpage by clicking on an advertisement sharing videos in online social network websites, are examples of progressive, i.e. irreversible actions. Nevertheless, there are numerous real world instances where the actions are *non-progressive* especially in technology adoption domain. For example, adopting a cell phone service provider, such as AT&T and T-mobile, is a non-progressive action where a user can switch between providers. The objective of influence maximization in this example is to persuade more users to adopt the intended provider for a longer period of time. Thus to gain more from the social influence of a costumer it is desirable not only make her purchase the product but also hold on to it for a long time. To capture the reversibility of choices in real scenarios, we present Heat Conduction (HC) model which has favorable real-world interpretation. We also show that HC unifies, generalizes, and extends the existing non-progressive models, including non-progressive LT (NLT) [24] and Voter model [67] (see Section 6.4). In contrast to the Voter model, HC does not *necessarily* reach consensus, where one product dominates and extinguishes the others after finite time, so the proposed HC model can explain the *coexistence of multiple product adoptions*, which is a typical phenomena in real world. In addition, HC model incorporates both "social" and "non-social" factors, e.g., intrinsic inertia or reluctance of some users in adopting a new idea or trying out a new product, external "media effect" which exerts a "non-social" influence in promoting certain ideas or products.

We tackle the influence maximization problem under HC influence model with a *scalable* and provably *near-optimal* solution. Kempe et al.'s approach [24] for influence maximization under NLT model, is to reduce the model to (progressive) LT by replicating the network as many as time progresses and compute the influence spread by the same slow Monte Carlo method for the resulted huge network. This approach is practically impossible for large networks, specially for the *infinite time horizon*. We also prove that contrary to the Voter, for which the influence maximization can be solved *exactly* in polynomial time [67], the influence maximization for HC is NP-hard. We develop an approximation (greedy) algorithm for influence maximization under HC for infinite time horizon with guaranteed *near-optimal* performance. Exploiting probability theory and novel Markov chain metrics, we are able to provide *closed form* solution for both computing the influence spread and greedy selection step which entirely removes the need to explicitly evaluate each node as the best seed candidate; our fast and scalable algorithm, C2GREEDY, for influence maximization under HC removes the computational barrier that prevented the literature from considering the non-progressive influence models.

We show that the non-progressive influence maximization problem under our HC framework is NP-hard. However, unlike the progressive influence maximization problem considered in [24], we demonstrate that the non-progressive influence maximization problem under HC can be well approximated using a *scalable* and provable *near-optimal* algorithm. Our fast and efficient algorithm benefits from two key properties of the proposed HC framework, where we establish *closed-form* expressions for the influence function computation and the greedy seed selection. Through extensive experiments on several real and synthetic networks, we validate the efficacy of our algorithm and demonstrate that it outperforms the state-of-the-art methods.

Our extensive experiments on several and large real and synthetic networks validate the efficiency and effectiveness of our method which outperforms the state-of-the-art in terms of both influence spread and scalability. We show that the most influential nodes under progressive models not necessarily act as the most influentials under non-progressive models and a *designated* non-progressive algorithm is necessary. Moreover, we present the first real non-progressive cascade dataset which models the non-progressive propagation of research topics among network of researchers.

The rest of this chapter is organized as follows. First, we introduce our HC model in Section 6.1. Next, we show how to compute the influence spread for HC in closed form in Section 6.2. In Section 6.3, we present our efficient algorithm C2GREEDY for influence maximization under the HC model. Section 6.4 explains how HC unifies other non-progressive models and provides a more complete view of the HC model. Finally we conduct comprehensive experiments in Section 6.5 to illustrate performance of our algorithm.

6.1 Heat Conduction Influence Model

The heat conduction (HC) influence model is inspired by the resemblance of influence diffusion through a social network to heat conduction through an object, where heat is transferred from the part with higher temperature to the part with lower temperature. We provide a simple description of HC in this section and defer the complete view of it as well as its unification property to Section 6.4.

Considering directed graph $G = (\mathcal{V}, \mathcal{E})$ which represents the social (influence) network, the directed edge from node i to node j declares that i follows j (or equivalently j influences i). Edge weight ω_{ij} indicates the amount that i trusts j and unless specified $0 \leq \omega_{ij} \leq 1$. The set of i's neighbors, representing the nodes that influence i, is denoted by $\mathcal{N}(i)$. The influence cascade can be assumed as a *binary* process in which a node who adopts the "desired" product is called *active*, and *inactive* otherwise. Note that this assumption holds for the cases with multiple products as well, where the objective is to maximize the influence (publicity) of the "desired" product, and the rest are all considered "undesired". *Seed* is a node that has been selected for the direct marketing and remains active during the entire process. In HC model, the influence cascade is initiated from a set of seeds S and arbitrary values for other nodes. The *choice* of node i to become active or inactive at time $t + 1$ is a linear function of the choices of its neighbors at time t as well as its intrinsic (or non-social) bias toward activeness:

$$\mathbb{P}\big(\delta_i(t+1) = 1 | \mathcal{N}(i)\big) = \beta_i b + (1 - \beta_i) \sum_{j \in \mathcal{N}(i)} \omega_{ij} \delta_j(t), \tag{6.1}$$

where $\beta_i \in (0, 1)$, $b \in [0, 1]$, and $\sum_{j \in \mathcal{N}(i)} \omega_{ij} = 1$. Indicator function $\delta_i(t)$ is 1 when node i adopts the desired product at time t and 0 otherwise. We refer to (6.1) as the *choice rule*. The dependence on neighbors in (6.1) represents the "social" influence and the bias value b accounts for "non-social" influence which comes from any source out of the neighbors, e.g. media. The "non-social" influence can explain the cases where the "social" influence alone fails to model the cascades [107]. We discuss further interpretation and extensions of HC in Section 6.4.

Replacing the choice rule (6.1) in $\mathbb{P}\big(\delta_i(t+1)\big) = \sum \mathbb{P}(\delta_i(t+1) | \mathcal{N}(i)) \mathbb{P}(\mathcal{N}(i))$ results in the following *probabilistic* interpretation of the original binary HC model. Each node i has a value at time t denoted by $u(i, t)$ which represents the *probability* that she adopts the desired product at time t:

$$u(i, t+1) = \beta_i b + (1 - \beta_i) \sum_{j \in \mathcal{N}(i)} \omega_{ij} u(j, t), \tag{6.2}$$

Simple calculation shows that the bias value b can be integrated into the network by adding a bias node n (assuming that the network has $n-1$ nodes) with adoption probability b. Therefore, HC dynamics converts to the following:

$$u(i, t+1) = \sum_{j \in \mathcal{EN}(i)} \mathbf{P}_{ij} u(j, t), \tag{6.3}$$

where $\mathcal{EN}(i) = \mathcal{N}(i) \cup \{n\}$ is the extended neighborhood, $\mathbf{P}_{in} = \beta_i$, $u(n, t) = b$, and

$\forall j \neq n : \mathbf{P}_{ij} = (1 - \beta_i)\omega_{ij}$. Rewriting (6.3) in the following form shows that HC follows the discrete form of **Heat Equation** [108], which reveals the naming reason of HC influence model: $u(:, t+1) - u(:, t) = (\mathbf{P} - \mathbf{I})u(:, t)$, where $\mathbf{L} = \mathbf{I} - \mathbf{P}$ is the Laplacian matrix, $u(i, t)$ is the temperature of particle i at time t, and ":" denotes the vector of all entries.

6.2 HC Influence Spread

Influence spread of set $\mathcal{S}$ for time t is defined as the expected number of active nodes at time t of a cascade started with $\mathcal{S}$. Knowing that $u(i, t)$ is the probability of node i being active at time t, *influence spread* (or function) $\sigma(\mathcal{S}, t)$ is computed from:

$$\sigma(\mathcal{S}, t) = \sum_{i \in \mathcal{V}} u(i, t). \tag{6.4}$$

Motivated by the classical heat transfer methods, the initial and the boundary conditions should be specified to solve the heat equation and find $u(i, t)$ uniquely. In HC, the seeds $\mathcal{S}$ and the bias node are the boundary nodes and the rest are interiors. Assuming $\mathcal{S} = \{n-1, n-2, ..., n-|\mathcal{S}|\}$ and n as the bias node, HC is defined by the following heat equation system:

$$\begin{aligned}
\text{Main equation} &: \quad u(:, t+1) - u(:, t) = -\mathbf{L}u(:, t) \\
\text{Boundary conditions} &: \quad u(n, t) = b, \\
& \quad\;\; u(s, t) = 1 \quad \forall s \in \mathcal{S} \\
\text{Initial condition} &: \quad u(:, 0) = z + [0, ..., 0, \underbrace{1, ..., 1}_{|\mathcal{S}|}, b]',
\end{aligned} \tag{6.5}$$

where, as indicated in this formula, initial value $u(:, 0)$ is the sum of two vectors: the initial values of the interior nodes (z) and the initial values of boundaries (the second vector). The corresponding entries of boundaries in z are zero. In the continue, exploiting probability theory and novel Markov chain metrics, we provide a closed form solution to this heat equation system.

Social network G can be interpreted as an absorbing Markov chain where the absorbing states (boundary set $\mathcal{B}$) are the seeds and bias node, $\mathcal{B} = \mathcal{S} \cup \{n\}$, and $\mathbf{P}_{ij}$ is the probability of transition from i to j. The adoption probability of the nodes at time t, i.e. $u(:, t)$, can be written as a linear function of initial condition (6.3):

$$u(:, t) = \mathbf{P}^t u(:, 0), \tag{6.6}$$

where $\mathbf{P}$ is row-stochastic and has the following block form: $\mathbf{P} = \begin{bmatrix} \mathbf{R} & \mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$. The superscript indicates the time here. The boundary set by definition have fixed values over time and do not follow any other nodes which leads to the zero and identity blocks $\mathbf{I}_{(|\mathcal{S}|+1)\times(|\mathcal{S}|+1)}$. Blocks $\mathbf{R}$ and $\mathbf{B}$ represent transition probabilities of interior-to-interior and interior-to-boundary respectively. Note that different boundary conditions in (6.5), like different seed set, result in a different $\mathbf{P}$. Therefore both $\mathbf{P}$ and $u(:,t)$ implicitly depend on $\mathcal{S}$.

When t goes to infinity, transient part of u vanishes and it converges to the steady-state solution $\mathbf{v} = u(:,\infty)$, which is independent of time and is Harmonic, meaning that it satisfies $\mathbf{Pv} = \mathbf{v}$ [109]. Assume $\mathbf{v} = \big(\mathbf{v}_{\mathcal{I}}, \mathbf{v}_{\mathcal{B}}\big)^T$ where $\mathcal{I} = \mathcal{V} \setminus \mathcal{B}$ is the set of interior nodes, then the value of interior nodes is computed from boundary nodes [109]:

$$\mathbf{v}_{\mathcal{I}} = (\mathbf{I} - \mathbf{R})^{-1}\mathbf{B}\mathbf{v}_{\mathcal{B}} = \mathbf{F}\mathbf{B}\mathbf{v}_{\mathcal{B}} = \mathbf{Q}\mathbf{v}_{\mathcal{B}}. \tag{6.7}$$

where $\mathbf{F} = (\mathbf{I} - \mathbf{R})^{-1}$ is the *fundamental matrix* and $\mathbf{F}_{ij}$ indicates the average number of times that a random walk started from i passes j before absorption by any absorbing (boundary) nodes [109]. Also, the *absorption probability* matrix $\mathbf{Q} = \mathbf{FB}$ is a $(n - |\mathcal{S}| - 1) \times (|\mathcal{S}| + 1)$ row-stochastic matrix, where $\mathbf{Q}_{ij}$ denotes the probability of absorption of a random walk started from i by the absorbing node j [109].

From here on, without loss of generality, we assume b to be zero in equation (6.5). Using (6.6) and (6.7), the influence spreads for infinite time can be computed in closed form:

$$\sigma(\mathcal{S},\infty) = \sum_{i=1}^{n} \mathbf{v}(i) = |\mathcal{S}| + \sum_{i\in\mathcal{I}} \sum_{s\in\mathcal{S}} \mathbf{Q}^{\mathcal{S}}_{is}. \tag{6.8}$$

The superscript in $\mathbf{Q}^{\mathcal{S}}$ and $\mathbf{P}^{\mathcal{S}}$ explicitly indicates that they are functions of seed set $\mathcal{S}$. Note that in fact they are depending on the total boundary set, $\mathcal{B} = \mathcal{S} \cup \{n\}$, but since the bias node is always a boundary, throughout this paper we discard it from the superscripts to avoid clutter.

6.3 Influence Maximization for HC

In this section we solve the influence maximization problem for *infinite time horizon* under HC model, formulated as follows:

$$\mathcal{S}^* = \underset{\mathcal{S}\subseteq\mathcal{V}}{\operatorname{argmax}}\, \sigma(\mathcal{S},\infty), \qquad s.t. \qquad |\mathcal{S}| \leq K. \tag{6.9}$$

6.3.1 Influence maximization for $K = 1$

Based on (6.8) and (6.9), the most influential person (MIP) is the solution of the following optimization problem: $\operatorname{argmax}_{\mathcal{V}\setminus\{n\}} \sum_{i\in\mathcal{V}\setminus\{s,n\}} \mathbf{Q}_{is}^{\{s\}}$. This equation states that to find the MIP, we need to pick each candidate s and make it absorbing and compute the new $\mathbf{P}$ as $\mathbf{P}^{\{s\}}$ which in turn changes $\mathbf{Q}$ to $\mathbf{Q}^{\{s\}}$, and repeat this procedure $n-1$ times for all s. This procedure is problematic because for each $\mathbf{Q}^{\{s\}}$ we require to recompute matrix $\mathbf{F}^{\{s\}}$ which involves matrix inversion. But, in the following theorem we show that we are able to do this by only one matrix inversion instead of $n-1$ matrix inversions, and having matrix $\mathbf{F}^{\emptyset}$ is enough to find the most influential person of the network ($\emptyset$ sign indicated no seed is selected):

Theorem 6.1 *MIP under HC (6.1) when $t \to \infty$ can be computed in closed form from the following formula:*

$$MIP = \operatorname*{argmax}_{s\in\mathcal{V}\setminus\{n\}} \sum_{i\in\mathcal{V}\setminus\{n\}} \frac{\mathbf{F}_{is}^{\emptyset}}{\mathbf{F}_{ss}^{\emptyset}} = \operatorname{argmax} \mathbf{1}'\breve{\mathbf{F}}^{\emptyset}, \tag{6.10}$$

where $\breve{\mathbf{F}}^{\emptyset}$ is $\mathbf{F}^{\emptyset}$ when each of its columns is normalized by the corresponding diagonal entry. Note that left multiplication of all ones row vector is just a column-sum operation.

6.3.2 Influence maximization for $K > 1$

Although the influence maximization can be solved optimally for $K = 1$, the general problem (6.9) under HC for $K > 1$ is NP-hard:

Theorem 6.2 *Given a network $G = (\mathcal{V}, \mathcal{E})$ and a seed set $\mathcal{S} \subseteq \mathcal{V}$, influence maximization for infinite time horizon (6.9) under HC defined by (6.1) is NP-hard.*

In spite of being NP-hard, we show that the influence spread $\sigma(\mathcal{S}, \infty)$ is *submodular* in the seed set $\mathcal{S}$ which enables us to find a provable near-optimal greedy solution. A set function $f : 2^{\mathcal{V}} \to \mathbb{R}$ maps subsets of a finite set $\mathcal{V}$ to the real numbers and is submodular if for $\mathcal{T} \subseteq \mathcal{S} \subseteq \mathcal{V}$ and $s \in \mathcal{V} \setminus \mathcal{S}$, $f(\mathcal{T} \cup \{s\}) - f(\mathcal{T}) \geq f(\mathcal{S} \cup \{s\}) - f(\mathcal{S})$ holds, which is the diminishing return property. Following theorem presents our established submodularity results.

Theorem 6.3 *Given a network $G = (\mathcal{V}, \mathcal{E})$, influence spread $\sigma(\mathcal{S}, \infty)$ under HC model is non-negative monotone submodular function.*

The greedy solution adds nodes to the seed set $\mathcal{S}$ sequentially and maximizes a monotone submodular function with $(1 - 1/e)$ factor approximation guarantee [110]. More formally the $(k+1)$-th seed is the node with maximum **marginal gain**:

$$(k+1)\text{th-}MIP_t = \underset{s \in \mathcal{V} \setminus \{\mathcal{S}_k \cup \{n\}\}}{\operatorname{argmax}} \; \sigma(\mathcal{S}_k \cup \{s\}, t) - \sigma(\mathcal{S}_k, t), \tag{6.11}$$

where $\mathcal{S}_k$ is the set of k seeds which have been picked already. Although we can compute the above objective function in closed form, for selecting the next seed we have to test all s to solve the problem which is the approach of all existing greedy based method in the literature. Previously a lazy greedy scheme have been introduced to reduce the number testing candidate nodes s [59]. In the next section we go one step further and show that under HC model and for *infinite time horizon* we can solve the marginal gain in *closed form*.

6.3.3 Greedy Selection

An important characteristic of the linear systems, like HC when $t \to \infty$, is the "superposition" principle. We leverage this principle to calculate the marginal gain of the nodes efficiently and pick the one with maximum gain for the greedy algorithm. Based on this principle, the value of each node in HC for infinite time, and for a given seed set $\mathcal{S}$, is equal to the algebraic sum of the values caused by each seed acting alone, while all other values of seeds have been kept zero. Therefore, when a node s is added to the seed set $\mathcal{S}_k$, its marginal gain can be calculated as the summation of values of the nodes when all of the values of $\mathcal{S}_k$ have been turned to zero and node s is the only seed in the network, whose value is $1 - \mathbf{v}^{\mathcal{S}_k}(s)$. In this new problem, the vector of boundary values $\mathbf{v}_{\mathbf{B}}^{\mathcal{S}_k \cup \{s\}}$ is a vector of all 0's except the entry corresponding to the node s with value $1 - \mathbf{v}^{\mathcal{S}_k}(s)$, and the value of interior node i is obtained from (6.7):

$$\mathbf{v}_{\mathcal{I}}^{\mathcal{S}_k \cup \{s\}}(i) = \mathbf{Q}_{is}^{\mathcal{S}_k \cup \{s\}}(1 - \mathbf{v}^{\mathcal{S}_k}(s))$$

Substituting $\mathbf{Q}$ from lemma 3 result (see Supplementary), the $k+1$-th seed is determined from the following closed form equation:

$$\begin{aligned} & (k+1)\text{th-}MIP \\ = \; & \underset{s \in \mathcal{V} \setminus \{\mathcal{S}_k \cup \{n\}\}}{\operatorname{argmax}} \sum_{i \in \mathcal{V} \setminus \{\mathcal{S}_k \cup \{n\}\}} \frac{\mathbf{F}_{is}^{\mathcal{S}_k}}{\mathbf{F}_{ss}^{\mathcal{S}_k}} \big(1 - \mathbf{v}^{\mathcal{S}_k}(s)\big), \\ = \; & \operatorname{argmax}(\mathbf{1} - \mathbf{v}^{\mathcal{S}_k})' \breve{\mathbf{F}}^{\mathcal{S}_k} \end{aligned} \tag{6.12}$$

Note that vector $\mathbf{v}^{\mathcal{S}_k}$ is obtained in step k and is known, and matrix $\mathbf{F}^{\mathcal{S}_k}$ can be calculated from $\mathbf{F}^{\mathcal{S}_{k-1}}$ without any need for matrix inversion (see Supplementary, lemma 1). One may observe that equation (6.12) is the general form of Theorem 1, since $\mathbf{v}^{\mathcal{S}_0} = \mathbf{v}^{\emptyset} = 0$. Notice that equation (6.12) intuitively uses two criteria for selecting the new seed: its current value should be far from 1 (higher value for $(1 - \mathbf{v}^{\mathcal{S}_k}(s))$ term) which suggests that it is far from the previously selected seeds, and at the same time it should have a high network centrality (corresponding to the $\mathbf{F}_{is}^{\mathcal{S}_k}/\mathbf{F}_{ss}^{\mathcal{S}_k}$ term). Algorithm 3 summarizes our C2GREEDY method for $t \to \infty$: a greedy algorithm with 2 closed form steps. Operator $\otimes$ in step 10 denotes the Hadamard product.

Algorithm 3 C2GREEDY

```
1:  input: extended directed network $G = (\mathcal{V}, \mathcal{E})$ with bias node $n$, maximum budget $K$.
2:  output: seed set $\mathcal{S}_K \subseteq \mathcal{V}$ with cardinality $K$.
3:  compute matrix $\mathbf{P}$ from $G$.
4:  $\mathcal{S}_0 := \emptyset$
5:  $\mathbf{F}^{\mathcal{S}_0} := (\mathbf{I} - \mathbf{P}^{\mathcal{S}_0})^{-1}$
6:  $s = \text{argmax}\, \mathbf{1}'\breve{\mathbf{F}}^{\emptyset}$, and $\mathcal{S}_1 = \mathcal{S}_0 \cup \{s\}$
7:  $\mathbf{v}^{\mathcal{S}_1} = \breve{\mathbf{F}}^{\mathcal{S}_0}(:, s)$
8:  for $k = 1$ to $K - 1$ do
9:      $\forall i, j \in \mathcal{I} : \mathbf{F}_{ij}^{\mathcal{S}_k \cup \{s\}} = \mathbf{F}_{ij}^{\mathcal{S}_k} - \frac{\mathbf{F}_{is}^{\mathcal{S}_k}\mathbf{F}_{sj}^{\mathcal{S}_k}}{\mathbf{F}_{ss}^{\mathcal{S}_k}}$
10:     $s = \text{argmax}(\mathbf{1} - \mathbf{v}^{\mathcal{S}_k})' \otimes \mathbf{1}'\breve{\mathbf{F}}^{\mathcal{S}_k}$
11:     $\mathcal{S}_{k+1} = \mathcal{S}_k \cup \{s\}$
12:     $\mathbf{v}^{\mathcal{S}_{k+1}} = \mathbf{v}^{\mathcal{S}_k} + (\mathbf{1} - \mathbf{v}^{\mathcal{S}_k}(s))\breve{\mathbf{F}}^{\mathcal{S}_k}(:, s)$
13: end for
```

6.4 Discussion

In this section, we present the comprehensive view of HC model and elaborate its (unifying) relation to the other models by providing multiple interpretations.

Model	Non-Social influence	Weighted Edges	Boundary		Init. Cond.		Eq. Physical Heat Conduction System
			H: $T = 1$	L: $T < 1$	$= 0$	$\neq 0$	
NLT1	√	√		√	√		Circular ring with a fixed-temp. point
NLT2	√	√	√	√	√		A rod with fixed-temp. ends, one high one low
NLT3		√			√		(Isolated) circular ring
NLT4		√	√		√		Circular ring with a fixed-temp. point
Voter						√	(Isolated) circular ring
GLT	√	√		√			Circular ring with a fixed-temp. point

Table 6.1: Specifying the equal heat system for existing non-progressive influence models.

6.4.1 Social interpretation

HC can be simply extended to model many real cases that the other influence models fail to cover. As briefly mentioned in Section 6.1, the original HC (6.1), models both "social" and "non-social" influences which cover the observations from the real datasets [107]. The extension of HC which is more flexible in modeling real world cascades is as follows:

$$u(i, t+1) = m\alpha_i + r\gamma_i + (1 - \gamma_i - \alpha_i) \sum_{j \in \mathcal{N}(i)} \omega_{ij} u(j, t), \tag{6.13}$$

where, $\sum_{j \in \mathcal{N}(i)} \omega_{ij} = 1$, $\gamma_i, \alpha_i \in [0, 1]$, $m = 1$, and $r = 0$. Factor r models the "discouraging" factor like intrinsic *reluctance* of customers toward a new product, and m represents "encouraging" factor like *media* that promotes the new product. These two factors can explain cases where all neighbors of a node are active but the node remains inactive, or when a node becomes active while none of her neighbors are active [107]. Note that all of the formulas and results stated so far is simply applicable to the general HC model (6.13).

6.4.2 Unification of existing non-progressive models

HC (6.1) unifies and extends many of the existing non-progressive models. In the Voter model, a node updates its choice at each time step by picking one of its neighbors randomly and adopting its choice. In other words, the choice rule of node i is the ratio of the number of her active neighbors to her total number of neighbors. Thus, Voter's choice rule is the simplified form of

HC's choice rule (6.1) where ω_{ij} is equal to $\frac{1}{d_i}$ (d_i is the out-degree of node i) and all β_is are set to zero. Also, note that having $\beta_i = 0$ indicates that the Voter does not cover the "non-social" influence.

In the *non-progressive* LT (NLT) [24], each node is assigned a random threshold θ *at each time step* and becomes active if the weighted number of its active neighbors (at previous time step) becomes larger than its threshold: $\sum_{j \in \mathcal{N}(i)} \omega_{ij} \delta_j(t) \geq \theta_i(t+1)$, where the edge weights satisfy $\sum_{j \in \mathcal{N}(i)} \omega_{ij} \leq 1$. Thus, the choice rule of node i at time $(t+1)$ under the NLT is obtained from the following equation:

$$\begin{aligned} \mathbb{P}\big(\delta_i(t+1) = 1 | \mathcal{N}(i)\big) &= \mathbb{P}\big(\theta_i(t+1) \leq \Sigma \omega_{ij}^{\text{NLT}} \delta_j(t)\big) \\ &= \Sigma \omega_{ij}^{\text{NLT}} \delta_j(t), \end{aligned} \tag{6.14}$$

where the second equality is the result of sampling $\theta_i(t+1)$ from the *uniform distribution* $U(0,1)$. Equation (6.14) is the simplified form of HC's choice rule (6.1), where $b = 0$ and $(1-\beta_i)\omega_{ij}^{\text{HC}} = \omega_{ij}^{\text{NLT}}$. Note that since in the NLT b accepts only zero value, this influence model also cannot cover *encouraging* "non-social" influence. Moreover, if the edge weights' gap in NLT, i.e. $g_i = 1 - \sum_{j \in \mathcal{N}(i)} \omega_{ij}^{\text{NLT}}$, is zero for all the nodes, it cannot model the "non-social" influence at all, since the corresponding β_i's in (6.1) would be equal to zero in that case.

Generalized linear threshold (GLT) is another non-progressive model proposed in [111] to model the adoption process of *multiple* products. Assigning a color $c \in \mathcal{C}$ to each product, a node updates its color, at each time step, by randomly picking one of its neighbors based on its edge weights and adopts the selected neighbor's color. For binary case $|\mathcal{C}| = 2$, where we only distinct between adoption of a desired product (active) and the rest of products (inactive), GLT's choice rule reduces to the following equation: $\mathbb{P}\big(\delta_i(t+1) = 1 | \mathcal{N}(i)\big) = \frac{\beta}{2} + (1-\beta) \sum_{j \in \mathcal{N}(i)} \omega_{ij} \delta_j(t)$. It is easy to see that this is the restricted form of HC's choice rule (6.1), where nodes are all connected to the bias node with equal weight of β and bias value b has to be $\frac{\beta}{2}$.

6.4.3 Physical interpretation

We showed that the existing non-progressive models are special cases of HC, and in this part we describe their equal heat conduction system which are uniquely specified by the initial and boundary conditions. Table 6.1 summarizes the heat interpretation of the influence models. We introduce four variants of non-progressive LT, based on two factors: seed and gap g_i. NLT1 and NLT2 support non-zero gaps, and NLT2 and NLT4 allows seeds, i.e. nodes in the network that always remain active. The non-progressive LT model presented in [24] is equivalent to NLT2. Reluctance factor and seeds in all models are equivalent to the low and high temperature boundaries respectively, and initial condition addresses the interiors' initial values (z in (6.5)). The non-social influence and edge weights factors appear in the Laplacian matrix calculation of (6.5). The equivalent physical heat conduction systems are easy to understand, here we just briefly point out the equivalence of the Voter model and the isolated circular ring. Circular ring is a rod whose ends are connected to each other and do not have any energy exchange with outside which explains why the Voter conserves the total initial heat energy, and reaches to an equilibrium with an equal temperature for all of the nodes, i.e., consensus.

6.4.4 Random walk interpretation

Beside the heat conduction view, the random walk prospect helps to gain a better understanding of the models and their relations. Assume that active and inactive nodes are colored black and white respectively. Consider the *original view* of any influence model which is the actual process that unfolds in time, so we look at the time-forward direction. We take a snapshot of the colored network at each time step t. Putting together the sequence of snapshots, the result is a random walk in the "colored graphs" state space with 2^n states. On the other hand, the *dual view* looks at the time-reverse direction of influence models. It is known for both IC-based models (like IC [24] and ConTinEst [64]) and LT-based models (Table 6.1 as well as HC and LT) that a single node from $\mathcal{N}(i)$ is responsible for i's color switch, which we name it as the parent of i. Now assuming that the process has advanced up to the time t, we reverse the process by starting from each node i and follow its ancestors. Here is the point where IC and LT based models separate from each other: due to $\sum_{j\in\mathcal{N}(i)} \omega_{ij} \leq 1$ constraint, ancestors of i in the LT-based models form a random walk starting from node i, which is not the case in IC-based models. Note that we have n random walks that can meet and merge, thus they are known as *coalescing random walks* [112]. This view also helps us to demonstrate the essential difference between progressive and non-progressive models. Dual view of progressive LT model is a *coalescing self-avoiding walks* which is the outcome of randomizing the threshold θ only once at the beginning of the process for the nodes in each realization. This bounds the number of "live" edges [24] connected to each node by one which prevents the creation of "loop" in the influence paths. Note that both counting and finding the probability of self-avoiding walks are

Table 6.2: List of networks used in experiments.

		$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	Params
Synthetic Networks	Random	1024	-	$[0.5, 0.5; 0.5, 0.5]$
	Hier.	1024	-	$[0.9, 0.1; 0.1; 0.9]$
	Core.	1024	-	$[0.9, 0.5; 0.5, 0.3]$
	ForestFire	1K-300K	$2.5\lvert\mathcal{V}\rvert$	$[0.35, 0.25]$
Real Networks	KClub	34	501	-
	PBlogs	1490	19087	-
	WikiVote	7115	103689	-
	MLWFW	10604	168918	-

$\#P$ hard [60].

6.5 Experiments

In this section, we examine several aspects of C2GREEDY and compare it with state-of-the-art methods. Experiments mainly focus on influence maximization and timing aspects. Finally, we present one example of real non-progressive data and illustrate the result of C2GREEDY.

6.5.1 Dataset

Table 6.2 summarizes the statistics of the networks that we use throughout the experiments. We work with both synthetic and real networks which we briefly discuss next.

Synthetic network generation. We consider the following types of Kronecker network for extensive performance comparison of our method with the state-of-the-art methods: random [113] (parameter matrix $[0.5, 0.5; 0.5, 0.5]$), hierarchical [114] ($[0.9, 0.1; 0.1; 0.9]$), and core-periphery [115] ($[0.9, 0.5; 0.5, 0.3]$). We generate 10 samples from each network and report the average performance of each method. Edge weights are drawn uniformly at random from $[0, 1]$ and weights of each node's outgoing edges is normalized to 1. For timing experiment, we use ForestFire [114] (Scale-free) network with forward and backward burning probability of 0.35 and 0.25, respectively, and set the outgoing edge weights of node i to $1/|\mathcal{N}(i)|$. The expected density, i.e., number of edges per node, for the resulted ForestFire networks is 2.5.

Real Networks. Zachary's karate club network (KClub) is a small friendship network with 34 nodes and 501 edges [116]. The political blogs network (PBlogs) [117], is a moderate size directed network of hyperlinks between weblogs on US politics with 1490 nodes and 19087 edges. Wikipedia vote network (WikiVote), is the network of who-vote-whom from wikipedia administrator elections [118] with 7115 nodes and 103689 edges. Finally, MLWFW is the network of who-follow-whom in the machine learning research community which we extract from citation networks of combined ACM and DBLP citation network which is available as a part of ArnetMiner [119]. For more information about MLWFW refer to Section 6.5.4.

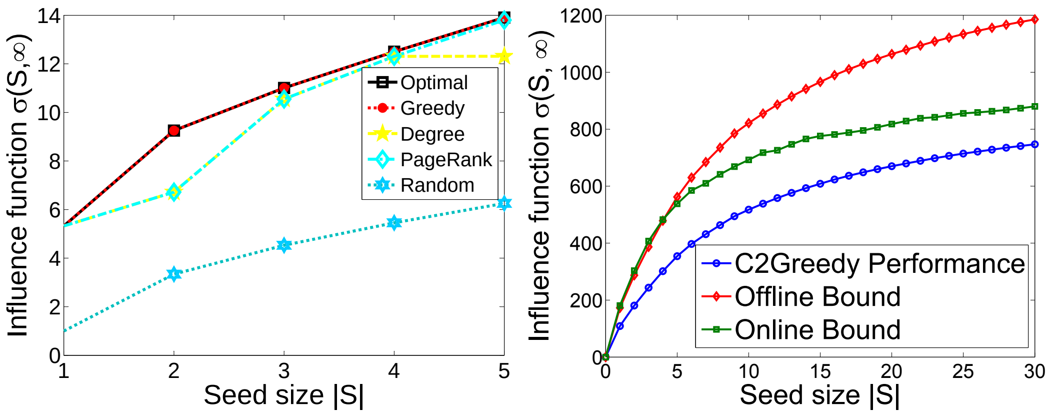
For all synthetic and real networks, after constructing the network, we add the bias node to the network and connect all nodes to it with weight $\beta_i = 0.1$ and re-normalize the weight of the other edges accordingly.

6.5.2 Influence Maximization

In this section we investigate the performance of C2GREEDY in the main task of influence maximization i.e., solving the set function optimization (6.9). Since finding the optimal solution for (6.9) is NP-hard, we compare C2GREEDY with optimal solution only for a small network, then for a large network we show that C2GREEDY result is close to the online bound [59]. We also compare the performance of C2GREEDY with the state-of-the-art methods proposed for solving (6.9) under different (mostly progressive) influence models.

C2GREEDY vs. optimal. For testing the quality of C2GREEDY method, we compare its performance with the best seed set (determined by brute force) on a small size network. We work with the KClub network for the brute-force experiment with $K = 5$. As Figure 6.1a shows C2GREEDY selects nodes that match the performance of the optimal seed set. In the next step, on a larger network, we show that the performance of C2GREEDY is close to the known online upper bound [59]. We compute the online and offline bounds of greedy influence maximization [59] with $K = 30$ for PBlogs network. Figure 6.1b illustrates that C2GREEDY result is close to the online bound and therefore close to the optimal solution's performance.

C2GREEDY vs. state-of-the-art. Next, we compare C2GREEDY with the state-of-the-art methods of influence maximization over three aforementioned synthetic networks and WikiVote real network. Among baseline methods PMIA [61] and LDAG [60] are approximation for IC and LT models respectively and SP1M [120] is a shortest-path based heuristic algorithm for influence maximization under IC. ConTinEst [121] is a recent method for solving continuous time model of [63] and PageRank is the well-known information retrieval algorithm [122]. Finally, Degree selects the nodes with highest degree as the most influential and Random picks



(a) C2Greedy vs. optimal.

(b) Online and offline bounds.

Figure 6.1: For small network (a) shows C2Greedy matches the optimal performance. For a larger network (b) compares performance of C2Greedy with online and offline bounds.

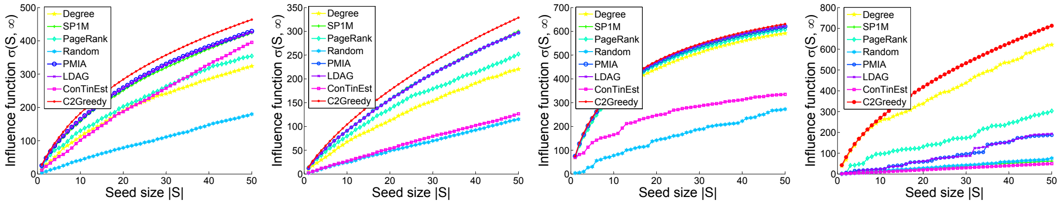
the seed set randomly.

The comparison results are depicted in Figure 6.2. Interestingly, our algorithm outperformed all of the baselines. Strangely, ConTinEst performs close to Random (except in the random network). A closer look at the results for three synthetic networks reveal that except ConTinEst's odd behavior all other methods have persistence rank in performance. C2GREEDY is the best method and is followed by PMIA and LDAG, both in second place, which are closely followed by SP1M. PageRank, Degree and Random are next methods in order. In WikiVote real network of Figure 6.2d surprisingly most of the state-of-the-art methods perform terribly poor and Degree (as the KMIP solution to Voter model) is the only competitor of C2GREEDY. Result of experiment with WikiVote shows that most influential nodes in a progressive models are not necessary influential in non-progressive ones, and designing non-progressive-specific algorithms (like C2GREEDY) is required for influence maximization under non-progressive models.

6.5.3 Speed and Scalability

In this part we illustrate the speed benefits of having two closed form updates in the greedy algorithm and also deal with the required single inverse computation of C2GREEDY to prove the scalability of our method.

Closed form benefits. As discussed in Section 6.3, our main algorithm C2GREEDY benefits from closed form computation for both influence spread (6.8) and greedy selection (6.12). To

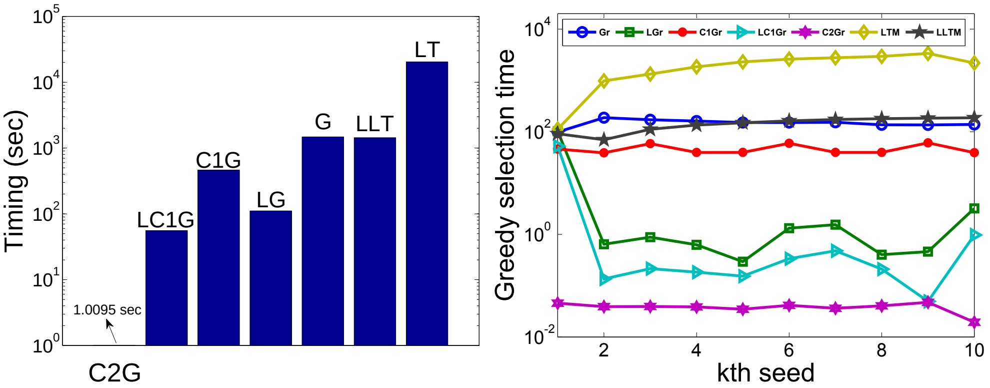


(a) Random network (b) Hierarchical network (c) Core-periphery network (d) Real network (WikiVote)

Figure 6.2: Comparing performance of C2Greedy with state-of-the-art influence maximization methods. Networks of (a), (b), and (c) are synthetic and (d) is a real network.

show the gain of these closed form solutions, we run the greedy algorithm in three different settings. First without using any of (6.8) and (6.12) which we call GREEDY and uses Monte Carlo simulation to estimate the influence spread. Second we only use (6.8) to have the closed form for influence spread without closed form greedy update of (6.12) which results in C1GREEDY, and finally C2GREEDY which uses both (6.8) and (6.12). Note that we can add lazy update of [59] (see Supplementary) to GREEDY and C1GREEDY to get LGREEDY and LC1GREEDY respectively. Finally we include the original greedy method [24] of solving LT model (progressive version of our model) and its lazy variant, with 100 iteration of Monte Carlo simulation. Note that for having a good approximation of influence spread in LT model, simulations are run for several thousand iterations, but here we just want to illustrate that the greedy algorithm for HC is much faster than LT, for which 100 iterations is enough. Figure 6.3a illustrates the speed in log-scale of all seven algorithms for $K = 10$ over the Pblogs dataset [117]. Note that the required time of inverse computation (6.7) is also included. The results confirm that both closed forms decrease the timing *significantly* (1 sec vs. 461 sec for the next best variation) and help the greedy algorithm far more than the lazy update.

Per-seed selection time. The major computational bottleneck of our algorithm is the inverse computation of (6.7). But fortunately this is needed once and at the beginning of the process. Here assuming offline inverse computation, we are interested in the cost of adding each seed. Figure 6.3b compares the cost of selecting k-th seed for the five variation of our algorithm, plus LT and LazyLT all described previously. As expected C2GREEDY requires the lowest computation time per seed. Also, the timing per seed for C2GREEDY is strictly decreasing over the size of $\mathcal{S}$, because the matrix N shrinks, while per seed selection time of LT is increasing



(a) Total time

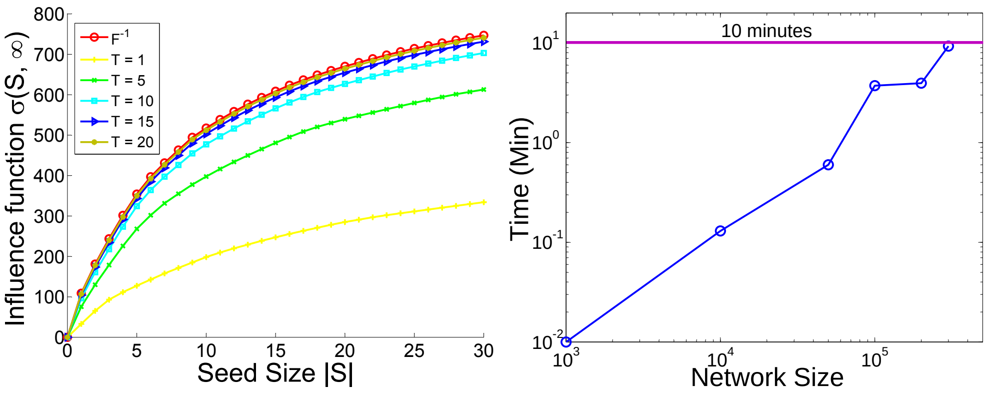
(b) Time per seed(sec)

Figure 6.3: In (a) we compare the total timing of seven algorithms to investigate the effect of closed updates on speed and in (b) we show the per-seed required time for the same experiment.

on average, because more seeds probably lead to bigger cascades.

Inverse approximation. Going beyond networks of size 10^4 makes the inverse computation problematic, but fortunately we have a good approximation of the inverse through the following expansion: $\mathbf{F} = (\mathbf{I}-\mathbf{R})^{-1} \approx \mathbf{I}+\mathbf{R}^1+\mathbf{R}^2+...+\mathbf{R}^T$. Since all eigenvalues of $\mathbf{R}$ are less than or equal to 1 contribution of $(\mathbf{R})^i$ to the summation drops very fast as i increases. The question is how many terms of the expansion, T, is enough for our application. Heuristically we choose the (effective) diameter of the graph as the number that provides us with a good approximation of $\mathbf{F}^{-1}$. Note that the ith term of the expansion pertains to the shortest paths of size i between any pair of nodes. Since the graph diameter is the longest shortest path between any pair of nodes, having that many terms gives us a good approximation of $\mathbf{F}^{-1}$. This is also demonstrated by the experimental result of Figure 6.4a where we compare the result of the influence maximization on the WikiVote network with diameter 15, with actual $\mathbf{F}^{-1}$ and its approximation for different T's. As discussed when T reaches to the diameter, the result of the algorithm that uses inverse approximation coincides with the algorithm that uses the exact inverse.

Scalability. Finally to show the scalability of C2GREEDY we perform influence maximization on networks with sizes up to 3×10^5. For speeding up the large scale matrix computation of the Algorithm 3 we developed an MPI version of our code which allows us to run C2GREEDY on computing clusters. Figure 6.4b shows the running time of C2GREEDY for ForestFire networks of sizes varying between 1K to 300K with edge density 2.5 (i.e. ratio of edges to nodes)



(a) Inverse Approximation

(b) Scalability for $K = 10$

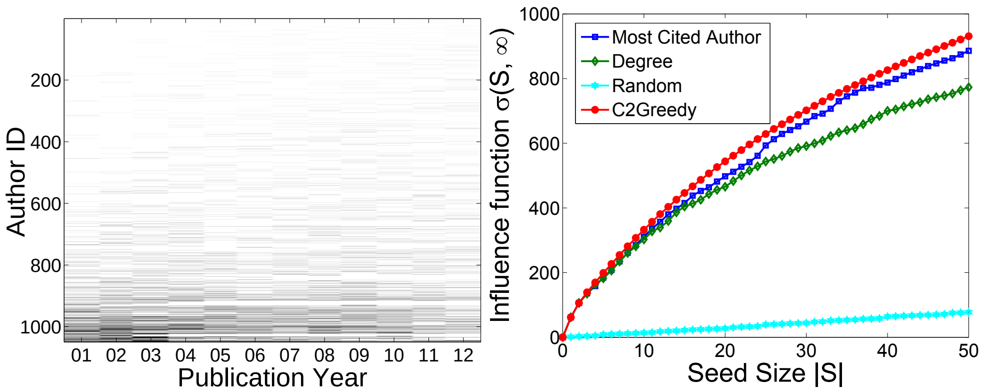
Figure 6.4: Timing for inf. max. in large scale networks by exploiting (a) inverse approximation and (b) parallel programming. Results of (b) are on FF networks with edge density 2.5.

and effective diameter of 10. The MPI code was run on up to 400 cores of 2.8 GHz. As Figure 6.4b indicates even for the largest tested network with 0.3 million nodes and 0.75 million edges C2GREEDY takes less than 10 minutes for $K = 10$.

To give a sense of our achievement in scalability we briefly mention the result of one of the state-of-the-art methods: The scalable ConTinEst [64] runs with 192 cores for almost 60 minutes on ForestFire network of size 100K and edge density of 1.5 to select 10 seeds, where our C2GREEDY finishes in less than 2 minutes for the similar ForestFire network (100K nodes and density 1.8) with 200 cores.

6.5.4 Real Non-progressive Cascade

Collaboration and citation networks are two well-known real networks that have been studied in social network analysis literature [24, 123]. Here we introduce a new network that represents who-follows-whom (WFW) in a research community. Note that the nodes in the collaboration and citation networks are authors and papers respectively but in WFW network nodes are authors and edges are inferred from citations. A directed edges (u, v) means that author u has cited one of the papers of author v which reveals that u follows/reads papers of v. Here we investigate the "research topic adoption" cascade. Researchers adopt new research topics during their careers and influence their peers along different research communities. The process starts with an arbitrary research topic for each author and they are influenced by the research topic



(a) Non-progressive cascade of ML research topic

(b) Inf. max. on inferred WFW network, ML-WFW.

Figure 6.5: In (a) we show the existence of non-progressive cascade of ML research topic where white means all papers of the author is about ML. In (b) we compare C2Greedy result with other baselines such as most cited author.

of those they follow and switch to another topic. For example a data mining researcher that follows mostly the papers of machine learning authors is probably going to switch his research topic to machine learning.

For illustration, we consider only the authors who have published papers in Machine Learning (ML) conferences and journals in a given time period. For the list of ML related conferences and journal we use resources of ArnetMiner project [119]. We consider each time step a year and study the years 2001 - 2012. An author is an *active* ML author in a given year if at least half of his publications in that year was published in ML venues. Figure 6.5a shows the change in the percentage of ML publication of ML authors who has more than 70 publication in years between 2001 and 2012. As Figure 6.5a suggests, cascade of ML research topic is a non-progressive process and researcher switch back and forth between ML and other alternatives. Among 1049 authors of Figure 6.5a about 400 of them are core ML authors who have rarely published in any other topic, but the non-progressive nature of the process is more visible in the rest (bottom part of Figure 6.5a).

Next we perform influence maximization on the inferred WFW network which we call MLWFW network. We extract the MLWFW network from the combined citation network of DBLP and ACM which is publicly available as a part of ArnetMiner project [119] and learn the edge weights similar to the weighted cascade model of [24]. The MLWFW network of 2001 - 2012 time frame consists of 10604 authors and 168918 edges. Figure 6.5b compares the result of influence maximization using C2GREEDY and other baselines. Note that other than regular baselines in this specific domain we have another well-known method which is "most cited author" that is equal to selecting authors with highest weighted in-degree in MLWFW network. As Figure 6.5b illustrates, C2GREEDY outperforms all of the other methods. Note that the list of K most influential authors in this experiment means that "if" those authors were switching to the ML topic completely (becoming a member of seed set $\mathcal{S}$) they would make the topic vastly popular. Therefore, although the seed set contains the familiar names of well-known ML authors (e.g., Michael I. Jordan and John Lafferty in first and second places), sometimes we encounter exceptions. For example, in the list of top 10 authors selected by C2GREEDY we have "Emery N. Brown" who is a renowned neuroscientist with publications in "Neural Computation" journal.

6.6 Proofs

6.6.1 Proof of Theorem 1

For proving this theorem we need the following lemmas.

Lemma 6.4 *When an interior node s is added to the current absorbing set $\mathcal{S}$, the new fundamental matrix $\mathbf{F}$ can be calculated from the previous one using the following equation:*

$$\mathbf{F}_{ij}^{\mathcal{S}\cup\{s\}} = \mathbf{F}_{ij}^{\mathcal{S}} - \frac{\mathbf{F}_{is}^{\mathcal{S}}\mathbf{F}_{sj}^{\mathcal{S}}}{\mathbf{F}_{ss}^{\mathcal{S}}},$$

Proof: The proof is straightforward based on Schur complement theorem [124]. This lemma helps avoiding the matrix inversion required for computing the new $\mathbf{F}^{\mathcal{S}\cup\{s\}}$ whenever an interior node s is added to the seed set $\mathcal{S}$.

Lemma 6.5 *The expected number of passages through an interior node and the expected number of passages through its interior neighbors has the following relation:*

$$\mathbf{F}_{ij}^{\mathcal{S}} = \begin{cases} \sum_k \mathbf{F}_{ik}^{\mathcal{S}}\mathbf{R}_{kj}^{\mathcal{S}} & i \neq j \\ 1 + \sum_k \mathbf{F}_{ik}^{\mathcal{S}}\mathbf{R}_{kj}^{\mathcal{S}} & i = j \end{cases}$$

Proof: We know $\mathbf{F}^{\mathcal{S}} = (\mathbf{I}-\mathbf{R}^{\mathcal{S}})^{-1}$. Start with $(\mathbf{I}-\mathbf{R}^{\mathcal{S}})^{-1}(\mathbf{I}-\mathbf{R}^{\mathcal{S}}) = I$ and after multiplication and rearranging we get to the lemma's statement: $\mathbf{F}^{\mathcal{S}} = \mathbf{I} + \mathbf{F}^{\mathcal{S}}\mathbf{R}^{\mathcal{S}}$

Lemma 6.6 *Starting from node i the absorption probability by node s, when $\mathcal{S} \cup \{s\}$ is the absorbing set, can be obtained from the expected number of passages through node s when it was not absorbing:*

$$\mathbf{Q}_{is}^{\mathcal{S}\cup\{s\}} = \frac{\mathbf{F}_{is}^{S}}{\mathbf{F}_{ss}^{S}}. \tag{6.15}$$

Proof:

$$\begin{aligned}
\mathbf{Q}_{is}^{\mathcal{S}\cup\{s\}} &= \sum_{j\in\mathcal{V}\setminus\{\mathcal{S}\cup\{s\}\}} \mathbf{F}_{ij}^{\mathcal{S}\cup\{s\}}\mathbf{B}_{js}^{\mathcal{S}\cup\{s\}} \\
&= \sum_{j\in\mathcal{V}\setminus\{\mathcal{S}\}} \mathbf{F}_{ij}^{\mathcal{S}\cup\{s\}}\mathbf{R}_{js}^{S} \\
&= \sum_{j\in\mathcal{V}\setminus\{\mathcal{S}\}} (\mathbf{F}_{ij}^{S} - \frac{\mathbf{F}_{is}^{S}\mathbf{F}_{sj}^{S}}{\mathbf{F}_{ss}^{S}})\mathbf{R}_{js}^{S} \\
&= \sum_{j\in\mathcal{V}\setminus\{\mathcal{S}\}} \mathbf{F}_{ij}^{S}\mathbf{R}_{js}^{S} - \frac{\mathbf{F}_{is}^{S}}{\mathbf{F}_{ss}^{S}} \sum_{j\in\mathcal{V}\setminus\{\mathcal{S}\}} \mathbf{F}_{sj}^{S}\mathbf{R}_{js}^{S} \\
&= \mathbf{F}_{is}^{S} - \frac{\mathbf{F}_{is}^{S}}{\mathbf{F}_{ss}^{S}}(\mathbf{F}_{ss}^{S} - 1) \\
&= \frac{\mathbf{F}_{is}^{S}}{\mathbf{F}_{ss}^{S}},
\end{aligned}$$

where the third and fifth equalities come from lemma 1 and lemma 2 respectively.

Proof of Theorem 1 is simply an instantiation of Lemma 3 for the case that we add node s as the first seed to the network and get $\mathbf{Q}_{is}^{\{s\}} = \frac{\mathbf{F}_{is}^{\emptyset}}{\mathbf{F}_{ss}^{\emptyset}}$, where $\emptyset$ emphasizes that the bias node is the only boundary. Note that all of the three lemmas are general in a sense that absorbing set can contain any type of boundary points, including zero-value node like the bias node and one-value node like a seed node.

6.6.2 Proof of Theorem 2

Proof: Consider an instance of the NP-complete Vertex Cover problem defined by an undirected and unweighted n-node graph $G = (\mathcal{V}, \mathcal{E})$ and an integer k; we want to know if there is a set $\mathcal{S}$ of k nodes in G so that every edge has at least one endpoint in $\mathcal{S}$. We show that this can be viewed as a special case of the influence maximization (6.9). Given an instance of the Vertex Cover problem involving a graph G, we define a corresponding instance of the influence maximization problem under HC for *infinite time horizon*, by considering the following settings in (6.1): (i) $\omega_{ij} = \omega_{ji} = 1$, if edge $(i - j) \in \mathcal{E}$, otherwise $\omega_{ij} = \omega_{ji} = 0$, (ii) bias node's value is zero $b = 0$, and (iii) β_i for all i's are equal to a known β. Note that since each interior node is connected to the zero-value bias node with edge weight β it cannot have value larger than $1 - \beta$. Hence, if there is a vertex cover $\mathcal{S}$ of size k in G, then one can deterministically make $\sigma(\mathcal{A}, \infty) = k + (n - k)(1 - \beta)$ by targeting the nodes in the set $\mathcal{A} = \mathcal{S}$; conversely, this is the only way to get a set $\mathcal{A}$ with $\sigma(\mathcal{A}, \infty) = k + (n - k)(1 - \beta)$.

6.6.3 Proof of Theorem 3

As mentioned in Section 6.3.3 when $t \to \infty$ superposition principle applies for HC model. We exploit this fact to prove the submodularity of influence spread. First note that $\sigma(\mathcal{S}, \infty)$ computed from (6.8) is the sum of node values and since the conic combination of submodular functions is also submodular it is enough to show that each node value, i.e., $\mathbf{v}(i)$ is submodular to proof Theorem 3. Here we need to work with the general set of bias nodes (compare to single bias node b) which we call ground set $\mathcal{G}$. We introduce a new notation where the value of node i is shown with $\mathbf{v}^{\mathcal{S},\mathcal{G}}(i)$. Also seed nodes can have arbitrary value of $\geq b$ instead of all 1 values.For proving the submodularity of $\mathbf{v}(i)$ we should prove:

$$\mathbf{v}^{\mathcal{T}\cup\{s\},\mathcal{G}}(i) - \mathbf{v}^{\mathcal{T},\mathcal{G}}(i) \geq \mathbf{v}^{\mathcal{S}\cup\{s\},\mathcal{G}}(i) - \mathbf{v}^{\mathcal{S},\mathcal{G}}(i), \mathcal{T} \subseteq \mathcal{S} \tag{6.16}$$

We invoke superposition to perform the subtraction:

$$\mathbf{v}^{\{s_{v_L}\},\mathcal{G}\cup\mathcal{T}}(i) \geq \mathbf{v}^{\{s_{v_R}\},\mathcal{G}\cup\mathcal{S}}(i), \qquad \mathcal{T} \subseteq \mathcal{S} \tag{6.17}$$

where v_L and v_R emphasize that the value of the new seed node is different in left and right hand side and is qual to $v_L = \big(1 - \mathbf{v}^{\mathcal{T},\mathcal{G}}(s)\big)$ and $v_R = \big(1 - \mathbf{v}^{\mathcal{S},\mathcal{G}}(s)\big)$. Note that $v_L \geq v_R$ since $\mathcal{T} \subseteq \mathcal{S}$. We can not compare the value of nodes in two different networks unless they share same grounds and seeds with possibly different values for each seed. Therefore, we try to make the grounds of both sides of (6.17) identical by expanding the LHS of (6.17) using superposition law [125]:

$$\mathbf{v}^{\{s_{v_L}\},\mathcal{G}\cup\mathcal{T}}(i) = \mathbf{v}^{\{s_{v_L}\},\mathcal{G}\cup\mathcal{S}}(i) + \mathbf{v}^{\mathcal{D},\mathcal{G}\cup\mathcal{S}\cup s,}(i) \tag{6.18}$$

where $\mathcal{D} = \mathcal{S} - \mathcal{T}$. Although second term of (6.18) is complicated but for our analysis it is enough to note that it is a non-negative number $\alpha \geq 0$. Now the submodularity inequality (6.16) reduces to:

$$\mathbf{v}^{\{s_{v_L}\},\mathcal{G}\cup\mathcal{S}}(i) + \alpha \geq \mathbf{v}^{\{s_{v_R}\},\mathcal{G}\cup\mathcal{S}}(i) \tag{6.19}$$

Now both sides have the same set of sources and grounds. Noticing that the value of the source in LHS is larger than RHS, i.e., $v_L \geq v_R$, and $\alpha \geq 0$ completes the proof.

Chapter 7

Conclusion

In this thesis, we presented our research in the domain of high dimensional problems in both discrete and continuous cases. For continuous problems, we focused on the structured linear regression where the structure is induced by a norm. In discrete problems, we studied a submodular maximization problem with cardinality constraint.

Chapter 3 presented a simple estimator for joint estimation of shared and private parameters of data sharing model. We show that the sample complexity of our estimator for estimation of the shared parameter depends on the total number of sample n. In addition, the shared parameter error rate decays as $1/\sqrt{n}$. These results indicate that our estimator really benefits from the pooled data in estimating the shared parameters. Both sample complexity and upper bound of error depend on the *maximum* Gaussian width among the spherical caps induced by the error cones of different parameters.

In Chapter 4, we investigated the consistency of the regularized estimators for structured estimation in high dimensional scaling when covariates are corrupted by additive sub-Gaussian noise. Our analysis holds for any norm $R(\cdot)$, and shows that when an estimate of the noise covariance is available, our estimators achieve consistent statistical recovery, and recently developed methods for sparse noisy regression are special cases. Finally, in the presence of additive noise, our method is stable, i.e., selects the correct support.

Chapter 5 moves to a more applied direction and uses tools from sparse coding and dictionary learning to perform tweet sentiment analysis. We presented a complete framework for tweet sentiment analysis in which we covered steps that should be preformed before any sentiment extraction. We formulated the sentiment analysis as three sequential 2-class classifications. In the first step, we separate tweets that are about the topic of interest and then filter out tweets that do not contain any emotion. Finally, we perform sentiment analysis on the resulted collection of tweets. Results of several classification algorithms were presented in both original space, i.e.,bag-of-words feature space and compressed space. Compression is performed using random reconstructible projection borrowed from compress sensing literature. Empirical results show that learning in compressed domain (compressed learning) is possible. Also, we presented a modification of all classifiers (i.e., NB, SVM, KNN and DL) that can deal with our weighted data label. Finally, we supplemented our per-tweet analysis with spatially aggregated results and showed that our approach also works well for batch-tweet analysis.

Lastly, in Chapter 6, we introduced the Heat Conduction Model which can capture both social influence and non-social influence, and extends many of the existing non-progressive models. We also presented a scalable and provably near-optimal solution for influence maximization problem by establishing three essential properties of HC: 1) submodularity of influence spread, 2) closed form computation for influence spread, and 3) closed form greedy selection. We conducted extensive experiments on networks with hundreds of thousands of nodes and close to million edges where our proposed method gets done in a few minutes, in sharp contrast with the existing methods. The experiments also certified that our method outperforms the state-of-the-art regarding both influence spread and scalability. Moreover, we exhibited the first real non-progressive cascade dataset for influence maximization. We believe that our method removes the computational barrier that prevented the literature from considering the non-progressive influence models.

References

[1] A. Asiaee T., M. Tepper, A. Banerjee, and G. Sapiro. If you are happy and you know it... tweet. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 1602–1606, 2012.

[2] Processing: What to record? , *https://home.cern/about/computing/processing-what-record.*

[3] L. Da Xu, W. He, and S. Li. Internet of things in industries: A survey. *IEEE Transactions on industrial informatics*, 10(4):2233–2243, 2014.

[4] Z. Dou, R. Song, and J. Wen. A large-scale evaluation and analysis of personalized search strategies. In *Proceedings of the 16th international conference on World Wide Web*, pages 581–590, 2007.

[5] A. G. Bosworth, C. Cox, R. Sanghvi, T. S. Ramakrishnan, and A. D'angelo. Generating a feed of stories personalized for members of a social network, 2010. US Patent 7,827,208.

[6] L. Wasserman. *All of statistics: a concise course in statistical inference*. Springer Science & Business Media, 2013.

[7] T. Hastie, R. Tibshirani, and M. J. Wainwright. *Statistical learning with sparsity: the lasso and generalizations*. CRC press, 2015.

[8] P. Buhlmann and S. van de Geer. *Statistics for High Dimensional Data: Methods, Theory and Applications*. Springer Series in Statistics. Springer, 2011.

[9] R. Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society*, 58(1):267–288, 1996.

[10] H. Zou and T. Hastie. Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society*, 67(2):301–320, 2005.

[11] E. Candes and T. Tao. The dantzig selector: statistical estimation when p is much larger than n. *The Annals of Statistics*, pages 2313 – 2351, 2007.

[12] S. N. Negahban, P. Ravikumar, M. J. Wainwright, and B. Yu. A Unified Framework for High-Dimensional Analysis of M-Estimators with Decomposable Regularizers. *Statistical Science*, 27(4):538–557, 2012.

[13] R. Vershynin. Estimation in high dimensions: a geometric perspective. In *Sampling theory, a renaissance*, pages 3–66. Springer, 2015.

[14] A. Banerjee, S. Chen, F. Fazayeli, and V. Sivakumar. Estimation with Norm Regularization. In *Advances in Neural Information Processing Systems*, pages 1556–1564. 2014.

[15] A. Asiaee T. and A. Banerjee. Structured high dimensional data sharing model. In *Submitted*, 2017.

[16] A. Asiaee T., S. Chaterjee, and A. Banerjee. High dimensional structured estimation with noisy designs. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pages 801–809, 2016.

[17] G. Golnari, A. Asiaee T., A. Banerjee, and Z. Zhang. Revisiting non-progressive influence models: Scalable influence maximization in social networks. In *Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence*, pages 316–325, 2015.

[18] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Structured sparsity through convex optimization. *Statistical Science*, 27(4):450–468, 2012.

[19] F. Bach. Learning with submodular functions: A convex optimization perspective. *Foundations and Trends® in Machine Learning*, 6(2-3):145–373, 2013.

[20] E. R. Elenberg, R. Khanna, A. G. Dimakis, and S. Negahban. Restricted strong convexity implies weak submodularity. *arXiv preprint arXiv:1612.00804*, 2016.

[21] S. M. Gross and R. Tibshirani. Data shared lasso: A novel tool to discover uplift. *Computational Statistics & Data Analysis*, 101:226–235, 2016.

[22] P. Loh and M. J. Wainwright. High-dimensional regression with noisy and missing data: Provable guarantees with nonconvexity. *The Annals of Statistics*, 40(3):1637–1664, 2012.

[23] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, 2010.

[24] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137 – 146, 2003.

[25] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. 2007.

[26] F. Dondelinger and S. Mukherjee. High-dimensional regression over disease subgroups. *arXiv preprint arXiv:1611.00953*, 2016.

[27] E. Ollier and V. Viallon. Joint estimation of k related regression models with simple l_1-norm penalties. *arXiv preprint arXiv:1411.1594*, 2014.

[28] E. Ollier and V. Viallon. Regression modeling on stratified data with the lasso. *arXiv preprint arXiv:1508.05476*, 2015.

[29] W. A. Fuller. *Measurement error models*. J. Wiley & Sons, 1987.

[30] A. Tumasjan, T. O Sprenger, P. G Sandner, and I. M Welpe. Predicting elections with Twitter: what 140 characters reveal about political sentiment. In *Proceedings of 4th International AAAI Conference on Weblogs and Social Media*, 2010.

[31] J. Bollen, H. Mao, and X.-J. Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, 2011.

[32] J. Bollen, A. Pepe, and H. Mao. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. In *Proceedings of 5th International AAAI Conference on Weblogs and Social Media*, 2011.

[33] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau. Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media*, pages 30–38, 2011.

[34] S. Golder and M. Macy. Diurnal and seasonal mood vary with work, sleep, and daylength across diverse cultures. *Science*, 333(6051):1878–1881, 2011.

[35] P. Dodds, K. Harris, I. Kloumann, C. Bliss, and C. Danforth. Temporal patterns of happiness and information in a global social network: hedonometrics and Twitter. *PLoS ONE*, 6(12):e26752, 2011.

[36] A. Krause and D. Golovin. Submodular function maximization., 2014.

[37] A. Krause, J. Leskovec, C. Guestrin, J. VanBriesen, and C. Faloutsos. Efficient sensor placement optimization for securing large water distribution networks. *Journal of Water Resources Planning and Management*, 134(6):516–526, 2008.

[38] A. Krause and C. E. Guestrin. Near-optimal nonmyopic value of information in graphical models. *arXiv preprint arXiv:1207.1394*, 2012.

[39] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 61 – 70, 2002.

[40] A. Bruckstein, D. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51(1):34–81, 2009.

[41] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 689–696, 2009.

[42] V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. S. Willsky. The convex geometry of linear inverse problems. *Foundations of Computational Mathematics*, 12(6):805–849, 2012.

[43] M. J. Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using l1-constrained quadratic programmming (Lasso). *IEEE Transactions on Information Theory*, 55(5):2183–2201, 2009.

[44] R. K. Iyer and J. A. Bilmes. Submodular optimization with submodular cover and submodular knapsack constraints. In *Advances in Neural Information Processing Systems*, pages 2436–2444, 2013.

[45] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008.

[46] B. Liu. Sentiment analysis and subjectivity. In *Handbook of Natural Language Processing*. CRC Press, Taylor and Francis Group, 2nd edition, 2010.

[47] R. Vershynin. Introduction to the non-asymptotic analysis of random matrices. In *Compressed Sensing*, pages 210–268. Cambridge University Press, Cambridge, 2012.

[48] A. Argyriou, R. Foygel, and N. Srebro. Sparse prediction with the k-support norm. In *Advances in Neural Information Processing Systems*, pages 1457–1465, 2012.

[49] Y. Chen and C. Caramanis. Noisy and missing data regression: Distribution-oblivious support recovery. In *Proceedings of The 30th International Conference on Machine Learning*, pages 383–391, 2013.

[50] M. Rosenbaum and A. B. Tsybakov. Sparse recovery under matrix uncertainty. *The Annals of Statistics*, 38(5):2620–2651, 2010.

[51] A. Belloni, M. Rosenbaum, and A. B. Tsybakov. An $\{l_1, l_2, l_\infty\}$-regularization approach to high-dimensional errors-in-variables models. *Electronic Journal of Statistics*, 10(2):1729–1750, 2016.

[52] M. Rosenbaum and A. B. Tsybakov. Improved Matrix Uncertainty Selector. *arXiv:1112.4413*, 2011.

[53] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *Information Theory, IEEE Transactions on*, 53(12):4655–4666, 2007.

[54] E. Yang, A. Lozano, and P. Ravikumar. Elementary estimators for high-dimensional linear regression. In *Proceedings of the 31st International Conference on Machine Learning*, pages 388–396, 2014.

[55] E. Yang, A. C. Lozano, and P. K. Ravikumar. Elementary estimators for graphical models. In *Advances in Neural Information Processing Systems*, pages 2159–2167, 2014.

[56] S. Chatterjee, K. Steinhaeuser, A. Banerjee, S. Chatterjee, and A. Ganguly. Sparse group lasso: Consistency and climate applications. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pages 47–58, 2012.

[57] S. Chatterjee, S. Chen, and A. Banerjee. Generalized dantzig selector: Application to the k-support norm. In *Advances in Neural Information Processing Systems*, pages 1934–1942, 2014.

[58] I. Ramirez, P. Sprechmann, and G. Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3501–3508, 2010.

[59] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 420–429. ACM, 2007.

[60] W. Chen, Y. Yuan, and L. Zhang. Scalable influence maximization in social networks under the linear threshold model. In *IEEE 10th International Conference on Data Mining*, pages 88–97. IEEE, 2010.

[61] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1029–1038, 2010.

[62] A. Goyal, W. Lu, and L. V.S. Lakshmanan. Simpath: An efficient algorithm for influence maximization under the linear threshold model. In *IEEE 11th International Conference on Data Mining*, pages 211–220, 2011.

[63] M. Gomez-rodriguez, D. Balduzzi, and B. Schölkopf. Uncovering the temporal dynamics of diffusion networks. In *Proceedings of the 28th International Conference on Machine Learning*, 2011.

[64] N. Du, L. Song, M. Gomez Rodriguez, and H. Zha. Scalable influence estimation in continuous-time diffusion networks. In *Advances in neural information processing systems*, pages 3147–3155, 2013.

[65] P. Clifford and A. Sudbury. A model for spatial conflict. *Biometrika*, 60(3):581 – 588, 1973.

[66] R. Holley and T. Liggett. Ergodic theorems for weakly interacting infinite systems and the voter model. *The Annals of Probability*, 3(4):643 – 663, 1975.

[67] E. Even-Dar and A. Shapira. A note on maximizing the spread of influence in social networks. In *Web and Internet Economics*, pages 281 – 286, 2007.

[68] M. Rudelson and S. Zhou. Reconstruction from anisotropic random measurements. *IEEE Transactions on Information Theory*, 59(6):3434–3447, 2013.

[69] G. Raskutti, M. J. Wainwright, and B. Yu. Restricted eigenvalue properties for correlated gaussian designs. *Journal of Machine Learning Research*, 11:2241–2259, 2010.

[70] Q. Gu and A. Banerjee. High dimensional structured superposition models. In *Advances In Neural Information Processing Systems*, pages 3684–3692, 2016.

[71] E. Yang and P. Ravikumar. Dirty statistical models. In *Advances in Neural Information Processing Systems*, pages 611–619, 2013.

[72] A. Jalali, P. Ravikumar, S. Sanghavi, and C. Ruan. A Dirty Model for Multi-task Learning. In *Advances in Neural Information Processing Systems*, pages 964–972, 2010.

[73] M. B. McCoy and J. A. Tropp. The achievable performance of convex demixing. *arXiv preprint arXiv:1309.7478*, 2013.

[74] S. Oymak, B. Recht, and M. Soltanolkotabi. Sharp time–data tradeoffs for linear inverse problems. *arXiv preprint arXiv:1507.04793*, 2015.

[75] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient Projections onto the l1 -Ball for Learning in High Dimensions. In *Proceedings of the 25th international conference on Machine learning*, pages 272–279, 2008.

[76] J. P. Buonaccorsi. *Measurement error: models, methods, and applications*. CRC Press, 2010.

[77] P. J. Bickel, Y. Ritov, and A. B. Tsybakov. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, pages 1705 – 1732, 2009.

[78] M.J. Wainwright. Sharp Thresholds for High-Dimensional and Noisy Sparsity Recovery Using -Constrained Quadratic Programming (Lasso). *IEEE Transactions on Information Theory*, 55(5):2183 – 2202, 2009.

[79] P. Sprechmann, I. Ramirez, G. Sapiro, and Y. C. Eldar. C-hilasso: A collaborative hierarchical sparse modeling framework. *Signal Processing, IEEE Transactions on*, 59(9):4183–4198, 2011.

[80] S. Chen and A. Banerjee. Structured estimation with atomic norms: General bounds and applications. In *Advances in Neural Information Processing Systems*, pages 2890–2898, 2015.

[81] Y. Chen and C. Caramanis. Orthogonal matching pursuit with noisy and missing data: Low and high dimensional results. *arXiv preprint arXiv:1206.0823*, 2012.

[82] T. E. Nichols and A. P. Holmes. Nonparametric permutation tests for functional neuroimaging: a primer with examples. *Human brain mapping*, 15(1):1–25, 2002.

[83] M. Ojala and G. C. Garriga. Permutation tests for studying classifier performance. *The Journal of Machine Learning Research*, 11:1833–1863, 2010.

[84] B. OConnor, R. Balasubramanyan, B. Routledge, and N. Smith. From tweets to polls: Linking text sentiment to public opinion time series. In *Proceedings of 4th International AAAI Conference on Weblogs and Social Media*, 2010.

[85] E. Kouloumpis, T. Wilson, and J. Moore. Twitter sentiment analysis: The good the bad and the OMG! In *Proceedings of 5th International AAAI Conference on Weblogs and Social Media*, 2011.

[86] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 151–160, 2011.

[87] D. M. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[88] H. Shan, A. Banerjee, and N. Oza. Discriminative mixed-membership models. In *Proceedings of the 9th IEEE International Conference on Data Mining*, pages 466–475, 2009.

[89] L. Hong and B. Davison. Empirical study of topic modeling in twitter. In *Proceedings of the 1st Workshop on Social Media Analytics*, pages 80–88, 2010.

[90] W. Xin Zhao, J. Jiang, J. Weng, J. He, and E.-P. Lim. Comparing twitter and traditional media using topic models. In *European Conference on Information Retrieval*, pages 338–349, 2011.

[91] S. P. Kasiviswanathan, P. Melville, A. Banerjee, and V. Sindhwani. Emerging topic detection using dictionary learning. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 745–754, 2011.

[92] L. Barbosa and J. Feng. Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 36–44, 2010.

[93] A. Pak and P. Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *LREC*, 2010.

[94] Adam Bermingham and Alan F. Smeaton. Classifying sentiment in microblogs: is brevity an advantage? In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 1833–1836, 2010.

[95] L. Zhang, R. Ghosh, M. Dekhil, M. Hsu, and B. Liu. Combining lexicon-based and learning-based methods for twitter sentiment analysis. Technical report, HP Laboratories, 2011.

[96] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. Technical report, Stanford University, 2009.

[97] P. Welinder, S. Branson, S. Belongie, and P. Perona. The multidimensional wisdom of crowds. In *Advances in Neural Information Processing Systems*, pages 2424–2432. 2010.

[98] R. Calderbank, S. Jafarpour, and R. Schapire. Compressed learning: Universal sparse dimensionality reduction and learning in the measurement domain. Technical report, Rice University, 2009.

[99] D. Hsu, S. Kakade, J. Langford, and T. Zhang. Multi-label prediction via compressed sensing. In *Advances in Neural Information Processing Systems*. 2009.

[100] E. Bingham and H. Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the 7th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 245–250, 2001.

[101] D. Fradkin and D. Madigan. Experiments with random projections for machine learning. In *Proceedings of the 9th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 517–522, 2003.

[102] C. Cortes and V. Vapnik. Support-vector networks. In *Machine Learning*, pages 273–297, 1995.

[103] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–451, 2004.

[104] A. Y. Yang, S. S. Sastry, A. Ganesh, and Yi Ma. Fast $l1$-minimization algorithms and an application in robust face recognition: A review. In *IEEE 17th International Conference on Image Processing*, pages 1849–1852, 2010.

[105] Yang X., Song Q., and Y. Wang. Weighted support vector machine for data classification. In *International Journal of Pattern Recognition and Artificial Intelligence*, volume 2, pages 859– 864, 2005.

[106] P. Domingos and M. Richardson. Mining the network value of customers. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 57 – 66, 2001.

[107] Meeyoung Cha, Alan Mislove, and Krishna P. Gummadi. A measurement-driven analysis of information propagation in the flickr social network. In *Proceedings of the 18th International Conference on World Wide Web*, pages 721 – 730, 2009.

[108] G. F. Lawler. *Random walk and the heat equation*. 2010.

[109] P. G. Doyle and J. L. Snell. *Random walks and electric networks*. 1984.

[110] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265 – 294, 1978.

[111] Nishith Pathak, Arindam Banerjee, and Jaideep Srivastava. A generalized linear threshold model for multiple cascades. In *IEEE 10th International Conference on Data Mining*, pages 965 – 970, 2010.

[112] D. Aldous and J. Fill. Reversible markov chains and random walks on graphs, 2002.

[113] P. Erdős and A. Rényi. On the evolution of random graphs. In *Publication of the Mathematical Institute of the Hungarian Academy of Science*, pages 17– 61, 1960.

[114] A. Clauset, C. Moore, and M. E. J. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453:98–101, 2008.

[115] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani. Kronecker graphs: An approach to modeling networks. *Journal of Machine Learning Research*, 11:985–1042, 2010.

[116] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452 – 473, 1977.

[117] L. A. Adamic and N. Glance. The political blogosphere and the 2004 U.S. election: Divided they blog. In *Proceedings of the 3rd International Workshop on Link Discovery*, pages 36 – 43, 2005.

[118] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting positive and negative links in online social networks. In *Proceedings of the 19th International Conference on World Wide Web*, pages 641 – 650, 2010.

[119] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. ArnetMiner: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 990–998, 2008.

[120] Masahiro Kimura and Kazumi Saito. Tractable models for information diffusion in social networks. In *Principles and Practice of Knowledge Discovery in Databases*, pages 259–271. 2006.

[121] M. Gomez-Rodriguez and B. Schölkopf. Influence maximization in continuous time diffusion networks. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

[122] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. 1998.

[123] J. Tang, D. Zhang, and L. Yao. Social network extraction of academic researchers. In *IEEE 7th International Conference on Data Mining*, pages 292–301, 2007.

[124] F. Zhang. *The Schur complement and its applications*, volume 4. Springer, 2006.

[125] A. Agarwal and J. Lang. *Foundations of analog & digital electronic circuits*. 2005.