ADVANCES IN RELATIONSHIP CLUSTERING AND OUTLIER DETECTION

By

CHANG LIU

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Statistics

Written under the direction of

Rong Chen

And approved by

_____

_____

_____

_____

New Brunswick, New Jersey

January, 2021

ABSTRACT OF THE DISSERTATION

Advances in Relationship Clustering and Outlier Detection

by Chang Liu

Dessertation Director: Rong Chen

Abstract: Generalized linear models (GLMs) are very popular to solve response modeling problems. But GLM users often encounter the problem of over-dispersion if there exists unobserved heterogeneity within the data. The first topic of my dissertation mainly addresses this problem by introducing a clustering method: HSA (Heterogenous Sample Auto-grouping) method, to reveal the hidden structure and account for the unobserved heterogeneity for GLMs. Furthermore, we developed a modeling framework of applying HSA to recover the decision boundary controlled by some structural variable in GLMs. My second dissertation topic is about deriving a directed neighborhood-based approach for local outlier detection. With the prevalence of local outlier detection techniques like local outlier factor (LOF), local outlier detection draws more and more attention. Many outlier detection methods based on this concept give us an outlying score representing how likely the corresponding data object to be an outlier. But the interpretation of the score is not consistent across different data sets. In order to resolve this problem, we propose a local outlier detection approach: LoCO (Local COnnectivity) method. It has stable performance in some challenging scenarios compared with existing local outlier detection techniques.

An outline of the subsequent chapter content is given as follow:

Chapter 2 introduces a novel clustering method: HSA method. We formulate the prob-

lem with a convex objective function. Since solving the optimization function is not trivial due to the nonlinear loss and many penalty terms, we introduce IOSA (Iterative Operator-splitting for Samples Algorithm) to solve the problem. The convergence of the algorithm is theoretically proved. As to the theoretical analysis, we analyzed the minimax lower bound and prediction upper bound of this type of problems. In the end, we also provide numerical examples to validate the model performance. We apply HSA method onto a tourism data and a bank marketing data as well. The resulting groups are reasonably justified.

In Chapter 3, we introduce another application of HSA. HSA can be used to uncover the hidden structure within a data set. In many applications, the hidden structure of the data is actually determined by some structural variable which controls the general structure of the model instead of affecting the model as a standard covariate. We propose a three-stage modeling procedure: SD-HSA (Structural variable Driven-HSA) to solve such type of problems. At the first stage, we narrow down the structural variable candidates pool. Then we apply HSA incorporating structural variables information at the second stage. Finally, we select out the best model using model selection criteria like AIC or BIC. We also provide numerical and real data examples to explore the performance of the modeling framework.

Chapter 4 introduces a local outlier detection method: LoCO method. It quantifies the degree of outlyingness of each data subject by constructing a local asymmetric network (LAN). LoCO score is easy to interpret, and more robust to density changes compared with current existing local outlier detection methods like local outlier factor (LOF). Furthermore, we calculate the "p-value" of each data based on LoCO scores using conformal prediction technique. We compare the performance of LoCO method and LOF through series of simulation examples. We also apply the new method in real data in the end.

ACKNOWLEDGEMENTS

First I would like to express my deepest gratitude to my thesis advisor, Professor Rong Chen. Without his continuous guidance, encouragement and strong support, I could not have a chance to complete the dissertation. Professor Chen has lots of brilliant ideas. When I get stuck, he can always come with multiple resolutions to try on. He illustrated me to think and act like a researcher. I have benefitted a lot from his rigorous and professional work attitude, and I will keep following these spirits in my future career. I also want to thank Professor Yiyuan She. I still remember his great courses on machine learning when he visited Rutgers three years ago. He has great knowledge in lots of areas. He got me into the world of machine learning, and provided lots of valuable insights and ideas for my first research topic. I would never dive into this area so quickly without his help.

I wish to thank the faculties in the Department of Statistics for providing us a diversified and resourceful environment to explore all different kinds of possibilities. I am grateful to the graduate directors, Professor Crane and Professor Dasgupta, for their guidance and support for graduate study and life. Sincerely thanks to the former department chair, Professor Regina Liu, for her efforts to take care of each student.

I also want to say thanks to Professor Han Xiao, Professor Minge Xie for being my committee members and providing helpful comments on my dissertation.

Last but not least, I want to show my gratitude to my colleagues in the statistics program for sharing ideas and supporting each other. I had a very enjoyable time with them at the department.

Finally, I want to thank my family for their unconditional support and love all the time.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Data mining has become very popular for many years. Its goal is to extract information from any data source. From the problem itself, data mining can be classified into two areas: supervised learning and unsupervised learning. Supervised learning, which mainly includes regression or classification problems, uses information from a training data set including target values or class labels to find the prediction or classification rule to apply on to the test data set. Unsupervised learning, which includes clustering methods, in particular, is used in multivariate statistics to uncover latent groups suspected in the data or to discover groups of homogeneous observations. The aim thus is often defined as partitioning the data set such that the groups are as dissimilar as possible and that the observations within the same group are as similar as possible. The groups composing the partition are also referred to as clusters.

Clustering analysis and outlier detection are two important and related topics. They have widespread applications in both scientific and industrial fields. Under the fast development of computing resources, more complex and advanced techniques become possible to extract useful information from big data sets. Clustering analysis can be used for different purposes. First, it can be employed as an exploratory tool to detect structures in multivariate data sets and achieve a more parsimonious representation. Second, it can be used as prototypes quantisation and data compression. Third, latent group structures can be revealed to discover unobserved heterogeneity. Clustering is often referred to as an exploratory data analysis problem which aims at revealing interesting and useful grouping or formulation of the observations or features. However, specifying what is interesting or useful in a formal way is challenging. Hennig (2015) argues that the definition of true clusters depends on the context and the clustering aim. Thus, there may not exist a unique clustering

criterion or solution given the data, but different clustering aims imply different solutions and analysis should in general be aware of the ambiguity inherent in cluster analysis.

Since HSA is conducted under generalized linear model's framework, it can be viewed as a model-based clustering problem. Clustering under a model-based framework is both challenging and attractive. It is challenging because we need to cluster with both response variable and predictors. By properly clustering the observations into various groups, we could identify hidden structures or latent variables. In this way, one can fit separate models within each cluster or add additional features into the model to improve the performance. Finite mixture model (Frühwirth-Schnatter 2006a) is a very popular statistical modeling technique to handle heterogeneity issues. It relies on strong distributional assumptions, and would often require a complicated model selection. Our first research topic focuses on studying the response driven/assisted clustering under the generalized linear model's framework. It is formulated as an optimization problem. Thus, it does not have very strong distributional assumption compared with finite mixture models.

The occurrence of outliers can increase the difficulty of clustering analysis. A data set may contain a few anomalies that do not comply with the majority behavior or model of data. These extreme observations are often referred to as outliers. Outlier detection has always been an important problem ever since human starts to analyze data, and attracted an increasing attention in the machine learning, data mining and statistics literature. Practically it is a pervasive phenomenon in applications from credit cards, insurance or health care, intrusion detection for cyber-security, fault detection in safety critical systems, to military surveillance. Many methods have been proposed to detect outliers. Roughly speaking, they can be divided into three categories, neighborhood-based, subspace-based and ensemble-based methods. The neighborhood-based outlier detection methods mainly exploit the neighborhood information of a given data subject to determine whether it is far from its neighbors, or whether its density is small or not. The subspace-based detection methods identify anomalies by sifting through different features subsets in an ordered way.

The class of ensemble-based method combines the outputs of several detection algorithms or base detectors into a unified output by using integrated strategies. For this topic, we mainly study a local outlier detection problem which has drawn more and more attention recently. The existing techniques include the density-based approaches: local outlier factor (LOF) and its variations (Breunig et al. (2000), Tang et al. (2002), Lazarevic and Kumar (2005), Kriegel et al. (2009a), Kriegel et al. (2011)). These methods quantify the degree of outlying for each observation by calculating an outlying score. But the resulting score will have inconsistent interpretation across different data sets. Thus, our second research topic focuses on developing a directed neighborhood-based outlier detection method for local outlier detection.

The next section gives the motivation for the two research topics.

1.1 Motivation

1.1.1 Motivation for HSA

Data in the real world is not always perfect. Real-life data might follow a complex mixture distribution that cannot be realized to a simple one. Clustering is then useful to identify different subsets each of which may follow a distinct distribution. We can put the problem in a non-Gaussian setting due to the possible discreteness and over-dispersion of the data. For example, GLMs are widely used in statistical modeling, but the variability within the data may make a single GLM assumption unrealistic. In order to address this heterogeneity issue, we propose a method to auto-group the given samples into different groups within each standard GLM fits.

Notably the sample heterogeneity is quite common in real practice. It could be caused by missing covariates. For example, assuming we are fitting a linear regression for a given data set, and the true model contains both feature A (x_A) and a binary indicator feature B

(x_B) along with their interaction term (x_A and x_B):

$$y = \beta_A x_A + \beta_B x_B + \beta_{AB} x_A x_B + \epsilon,$$

where $\epsilon \sim N(0, \sigma^2)$ is an error term following normal distribution. If we are only provided with feature A when fitting the model. A model based on x_A only will not fit the data well. More concretely, due to the existence of the interaction term, the corresponding coefficient with respect to x_A becomes $\beta_A + \beta_{AB} x_B$. As a result, as x_B varies, the coefficient of x_A also varies accordingly. Thus, the changing pattern will be reflected from the value of both the response variable y and relevant features $\boldsymbol{X} = (x_A, x_B)$. On the other hand, if we can include sample clustering into the parameter estimation based on both y and $\boldsymbol{X}$, it could be used to describe the data with some prediction accuracy and one can even to recover B by identifying the sample groups.

Clustering analysis is an unsupervised learning method that constitutes a cornerstone of data analysis. It is not a recent invention, and there are different clustering methods. Generally, clustering algorithms can be categorized into hierarchical clustering methods, partitioning clustering methods, density-based clustering methods, grid-based clustering methods, and model-based clustering methods. All these methods have their own characteristics and can handle different types of clustering structure. Compared with those clustering methods, our auto-grouped estimation method: HSA (Heterogeneous Sample Auto-grouping) has its own advantages: (i). It is a clustering algorithm designed for predicting the response variable. Most clustering algorithms do not include the response variable. For example: k-means clustering or agglomerative clustering deal with a feature matrix only regardless of the response. Of course, clustering based on both response variable and relevant features brings nontrivial challenges. HSA is applicable to any data fitted by GLMs, which has a wide range of applications and potentiality. Clustering for dis-continuous data is usually more difficult compared with that for Gaussian numeric data. But in our

framework, any distribution is permitted. In this dissertation, we mainly use the most two common models: linear regression and logistic regression models to illustrate the idea and performance of our method. (ii). The predictive infrastructure assists in the computation and algorithm design for the proposed method. It owns unique solution. Thus, we will not suffer from problems of different initialization values leading to different clustering results, which are common for some partition-based clustering methods like k-means method. Furthermore, a lot of clustering methods have the difficulty of have to interpret and justify the obtained clusters, This problem might be alleviated when using our method. We can use the prediction accuracy to make judgement of the number of clusters and cluster sizes. All of these advantages make the HSA attractive. We will introduce the HSA in detail in chapter 2.

On the other hand, when we study HSA, we realize that one of the biggest challenges is that the model performance is closely related to the quality of some meta-parameters to control the weights in weighted l_2 penalty. But as we will show in Chapter 2, choosing q proper weight is difficult in general. Thus, we are motivated to think of another scenario of which the weight choosing process could take advantages. This results in the Structural variable Driven-HSA (SD-HSA) framework. The goal of the whole framework is to characterize different GLMs within a given data using a pre-chosen structural variable. This framework improves HSA from an unsupervised data exploration tool to a well-defined system for building a better model with the help of structural variables. As a result, we can incorporate more potentiality into HSA. In summary, SD-HSA framework assumes there exists some structural variable controlling the group structure. The structural variable would benefit the weight choosing process in this scenario. Once we get the estimated state labels for each data subject from HSA, we can build separate models based on the decision region of the structural variable. In this way, structural variables endow more power into the HSA.

SD-HSA modeling framework is powerful in the sense that it could be used to recover

linear and nonlinear decision boundaries flexibly based on the grouping results. It will give us more choices to construct generalized linear models. However, it is still a challenging problem, especially in practice, one often does not know the true structural variable. Thus, we propose a modeling framework from narrowing down structural variable candidates to building separate models using the the grouping structure obtained from the HSA. Chapter 3 will introduce the overall framework of the structural variable driven HSA process.

1.1.2 Motivation for Developing Local Connectivity Outlier

As the increasing need for efficient and effective analysis methods to make use of the information contained implicitly in the data, we not only need to find patterns applicable to a considerable portion of objects in a data set, but also want to find the exceptional cases in some scenarios. As a result, detecting exceptional cases gradually becomes an independent research area: outlier detection. Outlier detection has plenty of applications including detective criminal activities of various kinds (e.g. online payment fraud), rare events, deviation from the majority. Thus, finding such exceptions or outliers is another important topic that is worth to study about.

Outliers originally existed as the by-product of clustering algorithms. From the viewpoint of a clustering algorithm, outliers are objects not located in clusters of the data set. In every cluster each data is authorized with a degree of the membership (Behera, Ghosh, and Mishra 2012). The outlier is naturally detected in the clustering process. Various clustering approaches are used for the outlier detection (Angiulli and Fassetti 2007). These methods mainly focus on detecting global outliers. A few studies have also been conducted on outlier detection for large data sets (e.g. Knorr and Ng (1999), Arning, Agrawal, and Raghavan (1996), Knorr and Ng (1997), Knorr and Ng (1997)). While a more detailed discussion on these studies will be given in the next section, it suffices to point out here that most of these studies treat the outlier classification as a binary classification problem. Namely, either an object in the data set is an outlier or not. However, with the rapid development of

information technology, the structure of data sources is becoming more and more complex. Thus, it becomes more and more meaningful to assign to each object a degree of being an outlier. For example, we can assign a risk score of potential fraud activities. On the other hand, due to the instability of data collection and transmission technology, etc., the data sets obtained are often incomplete in terms of time and space. In this scenario, we only care about the outlier detection in a local scope. The data point is considered as an outlier if its value significantly deviated from the rest of the data points in the same context. The corresponding outliers obtained would be local or contextual outliers. Currently existing local outlier detection method like LOF (Local Outlier Factor) (Breunig et al. 2000) has some drawbacks. LOF generate an outlying score through the ratio of the average neighborhood densities and the density of the target data point. As we will show later (in Chapter 4), the quotient value is hard to interpret sometimes, data sets with changing densities might result in unreasonable scores. Our second research topic proposes a directed neighborhood-based approach that detects local outliers with with better properties than LOF.

Specifically, we introduce a new method for finding outliers in multi-dimensional data set. We introduce a LoCO (Local COnnectivity) score for each object in the data set, indicating its degree of outlyingness. The outlier factor is local in the sense that only a restricted neighborhood of each object is being considered. To the best of our knowledge, one of the most popular local outlier detection methods is LOF. Thus, we made series of simulation examples to compare the performance of our newly propose method and LOF. LoCO scores will be more robust to unexpected density changes within a data set, namely, when the densities vary, the corresponding LoCO score is more reasonable. Furthermore, we developed a conformal outlying score based on the LoCO score. It determines a p-value for each observation, and it measures the extent to which a classification label (degree of outlyingness) is consistent with other observations in the data. We will introduce the detail of the LoCO score in Chapter 4.

1.2 Literature Review

1.2.1 Literature on Clustering Methods

HSA is a clustering method. Before diving into the details about the model in the next chapter, we first give a review about some popular clustering methods:

- Hierarchical clustering creates a hierarchical decomposition of the objects. They are either agglomerative (bottom-up) or division (top-down). Different hierarchical methods are distinguished by the criterion for determining which two clusters to merge or split at each level. Agglomerative algorithms start with each object being a separate cluster itself, and then successively merge groups according to a distance measure like Euclidean distance etc.. Division algorithms follow the opposite strategy. They start with one group of all data subjects, and successively split groups into smaller ones until each object falls into one cluster, or as desired. The principle is Lance and Williams (1967) demonstrated that many agglomerative hierarchical methods are variations of a common recurrence formula (Cormack (1971), Everitt et al. (2011), GROVE (1984), Milligan (1979)). Some important articles include the introduction of Ward Jr (1963) minimum variance method and Johnson (1967)'s discussion of the complete and single link methods. D'Andrade (1978) introduced a routine based on the nonparametric U statistic. Ding and He (2002) introduced the merging and splitting process in hierarchical clustering method. They provide a comprehensive analysis of selection methods and propose several new methods that determine how to best select the next cluster for split or merge operation on cluster. CURE (Clustering Using REpresentatives) (Guha, Rastogi, and Shim 1998) is an agglomerative hierarchical clustering algorithm that creates a balance between centroid and all point approaches. It used a combination of random sampling and partitioning. BIRCH (Balanced Iterative Reducing and Clustering using Hierachies) (Zhang, Ramakrishnan, and Livny 1996) is an agglomerative hierarchical clustering algorithm

and especially suitable for very large databases. ROCK (RObust Clustering using linKs) (Guha, Rastogi, and Shim 1998) is a robust agglomerative hierarchical clustering algorithm based on the notion of links. It is also appropriate for handling large data sets. Linkage algorithms (Karypis, Han, and Kumar 1999) are agglomerative hierarchical methods that consider merging of clusters based on distance between clusters. There exist three linkage algorithms: Single-link (S-link), Average-link (Ave-link) and Complete link (Com-link). The quality of a pure hierarchical clustering method suffers from its inability to perform adjustment, once a merge or split decision has been executed. It will not be changed afterward. This merge or split might lead to somewhat misleading clusters if not well chosen at some step.

- A partitional clustering algorithm constructs partitions of the data. It assigns a set of data points into different clusters by using iterative processes based on a predefined criterion function. As a result, such algorithms has very high complexity. They are known as nonhierarchical clustering procedure because only a single data partition is produced (Anderberg (1973), Sneath (1977), Späth (1980)). The techniques range in complexity from Hartigan (1975)'s very simple leader algorithms to rather intricate iterative reallocation methods (Ball and Hall (1965), Friedman and Rubin (1967), and Wolfe (1970)). Some popular partitional clustering techniques include k-means clustering, fuzzy clustering and colored FCM. k-means clustering was first developed by MacQueen (1967). In k-means clustering, clusters are formed using Euclidean distance, and k classes are created to minimize the error function (Kutbay, Ural, and Hardalaç 2015). Fuzzy theory is firstly developed by Zadeh (1965) for defining adjustable degrees of memberships. Fuzzy theory creates intermediate sets rather than classical sets. In classical set, each data obbject is assigned into only one cluster. In contrast, data in fuzzy clusters can be represented in multiple clusters. This multiset assgnment can blong to all the clusters with a certain degree of membership (Bezdek 1973). This one object in multiset representation can be useful for sharply

separated cluster boundaries. The fuzzy C-Mean algorithm (FCM) is frequently used because of its ease of operation and reliability in many applications (Abdulshahed et al. (2015), Ji et al. (2014), Qiu et al. (2013), Kannan et al. (2012)). Colored image fuzzy C-Mean (C-FCM) involves color-based clustering using fuzzy sets. This 3D method is firstly given by Kutbay and Hardalaç (2017) as Robust Colored Image FCM (RCI-FCM). In partition clustering, determination of cluster size is important. This selection differs from data sets to data set. If data sets include more features to classify in a cluster, more clusters will be needed. But unfortunately, this cluster number is not known for many clustering problems. Generally experience give the cluster number. Estimation of the cluster number is one of the major problems for validation.

- Another type of popular clustering method is density based clustering method. Clusters may be treated as dense regions in the data space, where clusters are separated by a sparse region containing "relative few" data. Given this assumption, a cluster can either be of "regular" or "arbitrary" shape. The notion behind density based clustering is to detect clusters of non-spherical or arbitrary shapes. Some of the common density based clustering techniques are DBSCAN, OPTICS, VDBSCAN, DBCLASD, ST-DBSCAN and DENCLUE (Shah, Bhensdadia, and Ganatra (2012), Parimala, Lopez, and Senthilkumar (2011)). DBSCAN is short for density-based spatial clustering of applications with noise. It is one of the earliest density based clustering methods. Its key idea is that for any data point to belong to a cluster, there must be at least a given number of data points within a specified radius, namely, the density of the neighborhood around the data point should exceed a given threshold. A disadvantage of DBSCAN is that it struggles with data sets which contain clusters of varying densities (Ertöz, Steinbach, and Kumar 2003). VDBSCAN algorithm (varied density based spatial clustering of applications with noise) can detect clusters with varied density. Also, the method automatically selects several values of the input pa-

rameter *Eps* for different densities. Even, the parameter k is automatically generated based on the characteristics of the data set (Chowdhury, Mollah, and Rahman 2010). There are other variations of the DBSCAN algorithm. Detailed information about them can be found in Parimala, Lopez, and Senthilkumar (2011). OPTICS (order points to identify the clustering structure) algorithm is an indirect method. Namely, it does not explicitly produce a data set clustering. Instead, it outputs a cluster ordering. Objects in a denser cluster are listed closer to each other in the cluster ordering.

- Grid based clustering methods mainly focus on spatial data. It is aimed to quantize the data set into a number of cells and then work with data subjects belonging to these cells. As a result, they do not relocate points, but rather build several hierarchical levels of groups of objects. As a result, they are closer to hierarchical algorithms. But the merging in conducted on grids, and consequently clusters, does not depend on a distance measure. It is decided by a predefined parameter instead. The main advantage of grid based method is its fast processing time which depends on number of cells in each dimension in quantized space. Some popular grid based clustering methods include CLIQUE (CLustering in QUEst) (Agrawal et al. 1998), STING (STatistical INformation Grid) (Wang, Yang, and Muntz, n.d.), MAFIA (Merging of Adaptive Intervals Approach to Spatial Data Mining) (Goil, Nagesh, and Choudhary 1999), Wave Cluster (Sheikholeslami, Chatterjee, and Zhang 1998) and O-CLUSTER (Orthogonal partitioning CLUSTERing) (Milenova and Campos 2002).

- Model based clustering is another type of clustering method. traditional clustering methods like k-means clustering are based on the definition of similarities or dissimilarities between observations and groups of observations. These type of algorithms all belong to heuristic clustering. Model based clustering (mixture model) is different from heuristic clustering methods. It can help in the application of cluster analysis by requiring the analyst to formulate the probabilistic model which is used to fit the

data thus making the aims and the cluster shapes aimed for more explicit than what is generally the case if heuristic clustering methods are used. Mixture models for clustering is discussed in McLachlan and Peel (2004) and Frühwirth-Schnatter (2006b). In addition, several review articles on model-based clustering are available including Stahl and Sallis (2012) and McNicholas (2016).

Comparing to those clusterings methods in literatures, HSA has its own characteristics. First of all, HSA method has a similar flavor with partitional clustering methods like k-means clustering. Because it also partition the data set into different sub-groups based on the similarity between each data objects. On the other hand, it is not the same with traditional heuristic clustering method. Traditional heuristic clustering methods are not formulated under a model framework. Thus, they usually do not have a response variable (y). They only make clusters based on the similarities within some covariates (features) of interest ($\boldsymbol{X}$). As a result, its overall framework is simpler compared with the HSA. Furthermore, HSA method also endows some flavor from model based clustering method. The overall clustering problem is built under a concrete model framework. But compared with finite mixture models, HSA is free from distributional assumptions. We do not need to care two much about specifying the prior distribution under the probabilistic assumption compared with finite mixture model. In contrast, HSA is a relational based clustering algorithm. It is more flexible to use in real practice when specifying distributions for each feature within the data is infeasible.

1.2.2 Literature on Outlier Detection Methods

Local connectivity method is a local outlier detection technique. We will also review some popular outlier detection techniques based on the following three categories:

- Neighborhood-based detection: the basic idea is to identify outliers by virtue of the neighborhood information. Given a data object, the anomaly score is defined as the average distance (KNN (Ramaswamy, Rastogi, and Shim 2000)) or weighted

distance (KNNW (Angiulli and Pizzuti 2002)) to its k nearest neighbors. Another strategy is to take the local outlier factor (LOF (Breunig et al. 2000)) as the measurement of anomaly degree, in which the anomaly score was measured relative to its neighborhood. Based on LOF and LoOP (Kriegel et al. 2009a) provided for each data object an oulier probability as score, which is easily interpretable and can be compared over one data set. In ODIN (Oulier Detection using Indegree Numbers) (Hautamaki, Karkkainen, and Franti 2004), an object is defined as an outlier if it participates in at most T neighborhoods in kNN graph, where T is a control parameter.

- Subspace-based detection: Anomalies often exhibit unusual behaviors in one or more local or low-dimensional subspaces. The low-dimensional abnormal behaviors would be masked by full dimensional analysis (Aggarwal 2017). Zimek, Schubert, and Kriegel (2012) noted that for a data object in a high dimensional space, only a subset of relevant features offers valuable information, while the rest are irrelevant to the task. On the contrary, the existence of the irrelevant features might impede the separability of the anomaly detection model. As a result, subspace learning is a popular techniques to handle high-dimensional problems. Theses methods have two types of representations: the sparse subspace methods (Zhang et al. (2009), Dutta, Banerjee, and Reddy (2015), Zhang et al. (2014), Aggarwal and Philip (2005)) and relevant subspace methods (Kriegel et al. (2009b), Zhang et al. (2016), Muller et al. (2008), Müller, Schiffer, and Seidl (2010), Müller, Schiffer, and Seidl (2011)).

- Ensemble-based method: Ensemble learning is quite popular in machine learning (Zimek, Campello, and Sander (2014), Aggarwal and Sathe (2015)). It has a relatively better performance than other related techniques in many cases. Thus, ensemble learning is also frequently used for anomaly detection. The FB (Feature bagging) detection method (Lazarevic and Kumar 2005) is an outlier detection method frequently used in large, high dimensional data sets. It combines results from multiple

outlier detection algorithms that are applied using different set of features. Every outlier detection algorithm uses a small subset of features that randomly selected from the original feature set. Thus, each outlier detector identifies different outliers, and assigns to all data records outlier scores that corresponding to their probability of being outliers. The outlier scores computed by individual outlier detection algorithms are then combined to find better quality outliers. aims to detect anomalies using a scoring system that randomly selects subspaces. However, this results in irrelevant dimensions due to random subspace selection. There are several anomaly detection methods that consider both feature bagging and subsampling (Zimek et al. 2013). However, the variance of objects are difficult to obtain using feature bagging, and the final results tend to be sensitive to the size of subsampled data.

In summary, different types of outlier detection methods are proposed to handle different problems. How to formulate an outlier detection problem depends on the nature of the input data, types of outliers (global or local) and data labels. All of the outlier detection methods described above have their own advantages and disadvantages. Specifically, one of the neighborhood-based method: LOF is quite popular because of its convenience to use. But when there exist density changes within the data, LOF can not handle the problem quite well. In order to resolve it, we propose a directed neighborhood-based approach: LoCO to properly account for the neighborhood information of each data subject in the data set.

CHAPTER 2
HSA: CLUSTERING FOR GENERALIZED LINEAR MODELS

2.1 Introduction

The motivation of HSA (Heterogeneous Sample Auto-grouping) method derives from the demand to study a mixture model with a response (possibly discreet) available. By properly quantifying the pairwise similarities between each sample pair, our HSA method partitions a data set into a small number of groups to account for sample heterogeneity. Give a data set of n samples associated with a response $\boldsymbol{y} = (y_1, \ldots, y_n)^T \in \mathbb{R}^n$ and p features of interest $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^T \in \mathbb{R}^{n \times p}$, we assume

$$y_i \mid \boldsymbol{x}_i \sim GLM(\boldsymbol{\beta}^{[I_i]}), \quad 1 \leq i \leq n \tag{2.1}$$

where $I_i \in \{1, \ldots, J\}$ and $\boldsymbol{\beta}^{[I_i]}$ is a coefficient vector for the I_i-th model on the i-th sample. We can interpret I_i as J sub-populations assumed over the whole data. Each sub-population follows a GLM with however a possibly different set of parameters. The true label of I_i is the hidden structure that we want to recover. We will develop an efficient algorithm called IOSA (Iterative Operator-splitting for Samples Algorithm) to estimate I_i based on the given data. Since HSA obtains the grouping structure by solving an optimization problem with respect to a data dependent coefficient matrix, it treats the grouping and estimation problem as a whole.

There are many potential applications of HSA. In some areas, the aim is to find groups of observations with similar regression coefficients. For example, sales or marketing data collected on all customers may not have the right label to differentiate different groups of customers. We can use our model to separate the consumers base on, say price elasticities, to develop an optimal pricing policy for a market segment.

In some other areas of application like biology or medicine, we can use HSA to find out hidden confounders. For example, in a cross-sectional study to examine the relationship between a disease and other features of interest at a single point, there might exist missing or unobserved confounders that interact with the collected observed features to affect the response. As a result, the hidden confounder leads to dramatically different coefficients vector which cannot be uniform over all samples. Our method enables us to separate the data into different group under which we fit multiple GLMs and estimate the coefficients correspondingly to make a better prediction for the response.

The idea of HSA could also be used in community detection with some minor adjustments to its objective function. Identifying network communities can be viewed as the problem of clustering a set of nodes into communities, where a node can sometimes belong to multiple communities. Because all the nodes in each community share some common properties or attributes, and because they have many relationships among themselves, one could use such information to perform the clustering task. The first source is the data about the objects and their attributes. For example, the users' social network profiles, or authors' publication histories may tell us how to group. The second source of data comes from the edges connections between the objects like users from friendship, authors collaborate. Since the objective function of HSA is constructed by two parts: an overall loss and a regularization to enforce equi-sparsity, we can use the first data source to determine the loss function, and use the second data source to enforce the grouping stucture. In this way, we combine the network and attribute information into a single model, and they will jointly determine the group patterns.

HSA is related to convex clustering. Convex clustering ensures a unique global minimizer compared with other popular clustering methods like k-means or hierarchical clustering. Lindsten, Ohlsson, and Ljung (2011) and Hocking et al. (2011) formulated the clustering task as a convex optimization problem. Lindsten, Ohlsson, and Ljung (2011) considered several l_p norm penalty while Hocking et al. (2011) considered l_1, l_2 and l_∞ penalties.

Chi and Lange (2013) introduced two algorithmic frameworks to solve the convex clustering problem. Chen et al. (2015a) compared it with traditional hierarchical clustering. Similar to our model's setting, She (2010) also used a weighted l_2 penalty to enforce clustering. She 2009 grouped the predictors for generalized linear models using non-convex penalties including discrete l_0 and $l_0 + l_2$ type penalties. The newly proposed model applies convex relaxation under the generalized linear model's framework, and enables us to identify the hidden structure within the data and make the response modeling concurrently.

Finite-mixture models are also very popular to handle heterogeneity within the data. There is a large body of literature on finite mixture of regressions see, e.g. Frühwirth-Schnatter (2006a), in which each component distribution of a finite mixture is linked to a separate regression. An analyst can employ a (finite) mixture of many regressions model if heterogeneity is suspected in the relationship between covariates and the response variable. But there are too many possibilities of the latent subpopulations and may result in difficulties in model selection. More importantly, the methods based on the mixture model assumption require knowing the grouping information beforehand, as well as some parametric model assumptions, which might be unrealistic to obtain in lots of applications.

Compared to the aforementioned models in the literatures, HSA has its own characteristics and advantages. First of all, it gives a novel cluster generalized linear model that has a convex optimization formulation and admits simple and stable iterative algorithms guaranteed to converge to unique global minimizer. In this way, it is both flexible and easily to implement. Second, traditional clustering methods often suffer from choosing a proper number of clusters. The proposed method will generate a solution path from which we can often easily to visually recognize the right number of groups. Third, our algorithm is scalable and potentially apply to large-scale data. When computing resources are limited (for example, without GPU enabled environment), a model-based stratified sampling process can be used to reduce the computation burden further. In our experience, meaningful and robust grouping patterns can be generated effectively on several real data applications.

The rest of the chapter is organized as follow. We first specify the model assumptions along with its weighed regularizer. Then we will introduce the computational algorithms to solve the optimization problem. Next, we will make some non-asymptotic analysis on the minimax lower bound and upper error bound. In the end, simulation and real data examples will be provided.

2.2 Model Specification

2.2.1 Notations and Definitions

We summarize the notations and the definition of norms used in this chapter. Matrices are represented as boldface uppercase letters. Vectors are written as boldface lower case letters.

The l_p norm of any vector $\boldsymbol{v} \in \mathbb{R}^n$ is defined as $\|\boldsymbol{v}\|_p = (\sum_{i=1}^n |v_i|^p)^{\frac{1}{p}}$ $(0 \leq p \leq \frac{1}{2})$. The Frobenius norm of any matrix $\boldsymbol{M} = \{m_{ij}\} \in \mathbb{R}^{n \times r}$ is defined as

$$\|\boldsymbol{M}\|_F = \sqrt{\sum_{j=1}^{r} \sum_{i=1}^{n} m_{ij}^2}. \tag{2.2}$$

The l_p norm for $\boldsymbol{M}$ is defined as

$$\|\boldsymbol{M}\|_p = \sup_{\boldsymbol{x} \neq 0} \frac{\|\boldsymbol{M}\boldsymbol{x}\|_p}{\|\boldsymbol{x}\|_p}. \tag{2.3}$$

For $p, q \geq 1$, the $L_{p,q}$ norm is defined as

$$\|\boldsymbol{M}\|_{p,q} = \left(\sum_{j=1}^{r} \Big(\sum_{i=1}^{n} |m_{ij}|^p \Big)^{\frac{q}{p}} \right)^{\frac{1}{q}}. \tag{2.4}$$

Specifically, the $L_{2,\infty}$ norm is

$$\|\boldsymbol{M}\|_{2,\infty} = \max_{1 \leq i \leq n} \sqrt{\sum_{j=1}^{r} m_{ij}^2}. \tag{2.5}$$

Denote $\|\boldsymbol{M}\|_{2,C}$ to be the unique number of rows in $\boldsymbol{M}$. Furthermore, we define the l_2 inner product between two vectors with the same shape as

$$\langle \boldsymbol{v}_1, \boldsymbol{v}_2 \rangle_2 = \boldsymbol{v}_1^T \boldsymbol{v}_2. \tag{2.6}$$

For any matrix $\boldsymbol{N} = \{n_{ij}\} \in \mathbb{R}^{n \times r}$ with the same shape as $\boldsymbol{M}$, the Frobenius inner product between the two matrices is defined as

$$\langle \boldsymbol{M}, \boldsymbol{N} \rangle_F = \sum_{j=1}^{r} \sum_{i=1}^{n} m_{ij} n_{ij}. \tag{2.7}$$

2.2.2 The Statistical Model

Given a data set of n observations and p features denoted by $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}$ ($|\mathcal{D}| = n, \boldsymbol{x}_i \in \mathbb{R}^p, y_i \in \mathbb{R}$), a standard GLM response variable $Y_i = y_i$ is assumed as follows,

$$f(y_i \mid \boldsymbol{x}_i) \propto \exp(y_i \eta_i - \tau(\eta_i)), \quad \eta_i = \boldsymbol{x}_i \boldsymbol{\beta}, \tag{2.8}$$

where $f(y_i \mid \boldsymbol{x}_i)$ is the density given $\boldsymbol{x}_i$, $\boldsymbol{\beta} \in \mathbb{R}^p$ is a coefficient vector to be estimated. Equation (2.8) gives an example in the exponential family. Let the conditional mean $\mu_i = \mathbb{E}(Y_i \mid \boldsymbol{x}_i)$. Assuming

$$g(\mu_i) = \eta_i = \boldsymbol{x}_i \boldsymbol{\beta}, \tag{2.9}$$

where the *link function* $g(\cdot)$ specifies the relationship between the linear combination η_i. Some commonly used link functions include logit, log link, etc.. We also denote $\tau'(\cdot)$ as the derivative of the cumulant function $\tau(\cdot)$ in equation (2.8). The standard notation of GLM likelihood function often uses $b(\cdot)$, but to avoid confusion, $\tau(\cdot)$ is used thereafter.

The goal of HSA is to partition data $\mathcal{D}$ into J ($1 < J < n$) different groups such that the observations in the same group share approximately the same GLM parameter. Denote

the sample-dependent design matrix as $\boldsymbol{X}_i = (0, \ldots, \boldsymbol{x}_i, \ldots, 0)$ where only the i-th row is non-zero and equals $\boldsymbol{x}_i$. Assigning an independent coefficient $\boldsymbol{\beta}_i \in \mathbb{R}^p$ $(i = 1, \ldots, n)$ to each sample, equation (2.9) is replaced by:

$$g(\mu_i) = \boldsymbol{x}_i^T \boldsymbol{\beta}_i = \langle \boldsymbol{X}_i, \boldsymbol{B} \rangle_F. \tag{2.10}$$

Let $I_i \in \{1, \ldots, J\}$ be the membership variable where $1 \leq J \leq n$. Denote the concatenated coefficient matrix constructed by all coefficient vectors $\boldsymbol{\beta}_i$s as $\boldsymbol{B} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_n)^T \in \mathbb{R}^{n \times p}$. The $\boldsymbol{\beta}_i$s have only J distinct choices. Let $\boldsymbol{\beta}_j^*$ denote the coefficient vector for group j $(1 \leq j \leq J)$, then the group status of the i-th sample is determined by

$$\boldsymbol{\beta}_i = \boldsymbol{\beta}_j^* \quad \text{if } I_i = j.$$

As a result, we will estimate the group status of each sample by estimating the coefficient matrix $\boldsymbol{B}$ through an optimization problem. Next, we will specify the objective function to solve $\boldsymbol{B}$.

2.2.3 Regularization

Since each sample has its own coefficient vector $\boldsymbol{\beta}_i$, the systematic component defined in equation (2.8) should be $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_n)^T \in \mathbb{R}^n$ with $\eta_i = \boldsymbol{x}_i^T \boldsymbol{\beta}_i$. Denote the response variable as $\boldsymbol{y} \in \mathbb{R}^n$ and design matrix as $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^T \in \mathbb{R}^{n \times p}$. The objective function of HSA is

$$\begin{aligned} f(\boldsymbol{B} \mid \boldsymbol{y}, \boldsymbol{X}) &:= l(\boldsymbol{B} \mid \boldsymbol{y}, \boldsymbol{X}) + P(\boldsymbol{B}) \\ &= -\langle \boldsymbol{y}, \boldsymbol{\eta} \rangle_2 + \langle 1, \tau(\boldsymbol{\eta}) \rangle_2 + P(\boldsymbol{B}), \quad s.t.\ \eta_i = \langle \boldsymbol{X}_i, \boldsymbol{B} \rangle_F, \end{aligned} \tag{2.11}$$

where $l(\boldsymbol{B} \mid \boldsymbol{y}, \boldsymbol{X})$ is the GLM likelihood function defined in equation (2.8), $P(\boldsymbol{B})$ is a regularization term to enforce the equi-sparsity in $\boldsymbol{B}$ such that $\boldsymbol{B}$ will only have J unique rows. The general form of $P(\cdot)$ is $P(\boldsymbol{B}) = \sum_{1 \leq i < j \leq n} P(\boldsymbol{\beta}_i - \boldsymbol{\beta}_j; \lambda_{ij})$. The over-parametrization

form makes the regularization term $P(\cdot)$ play a crucial role. Fortunately, the subpopulation assumption means that $\boldsymbol{B}$ possesses a great deal parsimony, namely, the number of distinct rows in $\boldsymbol{B}$ is small. In order to enforce the equi-sparsity, we choose a weighted group l_1 form penalty:

$$P(\boldsymbol{B}) = \sum_{1 \le i < j \le n} P(\boldsymbol{\beta}_i - \boldsymbol{\beta}_j; \lambda_{ij}) = \lambda \sum_{1 \le i < j \le n} w_{ij} \|\boldsymbol{\beta}_i - \boldsymbol{\beta}_j\|_2, \tag{2.12}$$

where $\lambda_{ij} = \lambda w_{ij}$ is a data dependent weight to further improve the grouping pattern. Here w_{ij} is the weight to adjust the strength of the penalty on each difference term $\|\boldsymbol{\beta}_i - \boldsymbol{\beta}_j\|$, and λ is an overall regularization parameter. The penalty term improves the interpretability of the coefficient estimation. Similar to the rationale of LASSO (Tibshirani 2011) for variable selection, the pairwise penalty is able to make some rows in $\boldsymbol{B}$ exactly equal to each other. It is also worthy to note that the data dependent weight $\{w_{ij}\}_{(1 \le i < j \le n)}$ is indispensable based on the theoretical analysis in She (2009). A plain penalty $\lambda\|\boldsymbol{\beta}_i - \boldsymbol{\beta}_j\|_2$ would be weak and not effective in capturing the hidden structure. The weighted l_1 form or a non-convex sparsity-inducing penalty is more helpful.

With the help of the regularization term, we can generate a solution path of $\boldsymbol{B}$ by varying λ. Figure 2.1 shows an example of the solution path generated using the Tourism Data we would analyze later. X-axis stands for the total number of groups within the data. We choose 5 groups in total. Each group contains samples which lead to minor splits later.

2.2.4 Weighting Schemes

Since the choice of the weight $\{w_{ij}\}_{(1 \le i < j \le n)}$ is crucial to our model, in this section, we will introduce different weighting schemes.

Intuitively, if sample i and sample j are likely to belong to the same group, then the weight value w_{ij} should be sufficiently large to force $\|\boldsymbol{\beta}_i - \boldsymbol{\beta}_j\|_2 = 0$. On the contrary, if the grouping status between two samples is largely unknown, w_{ij} should not be overly large.

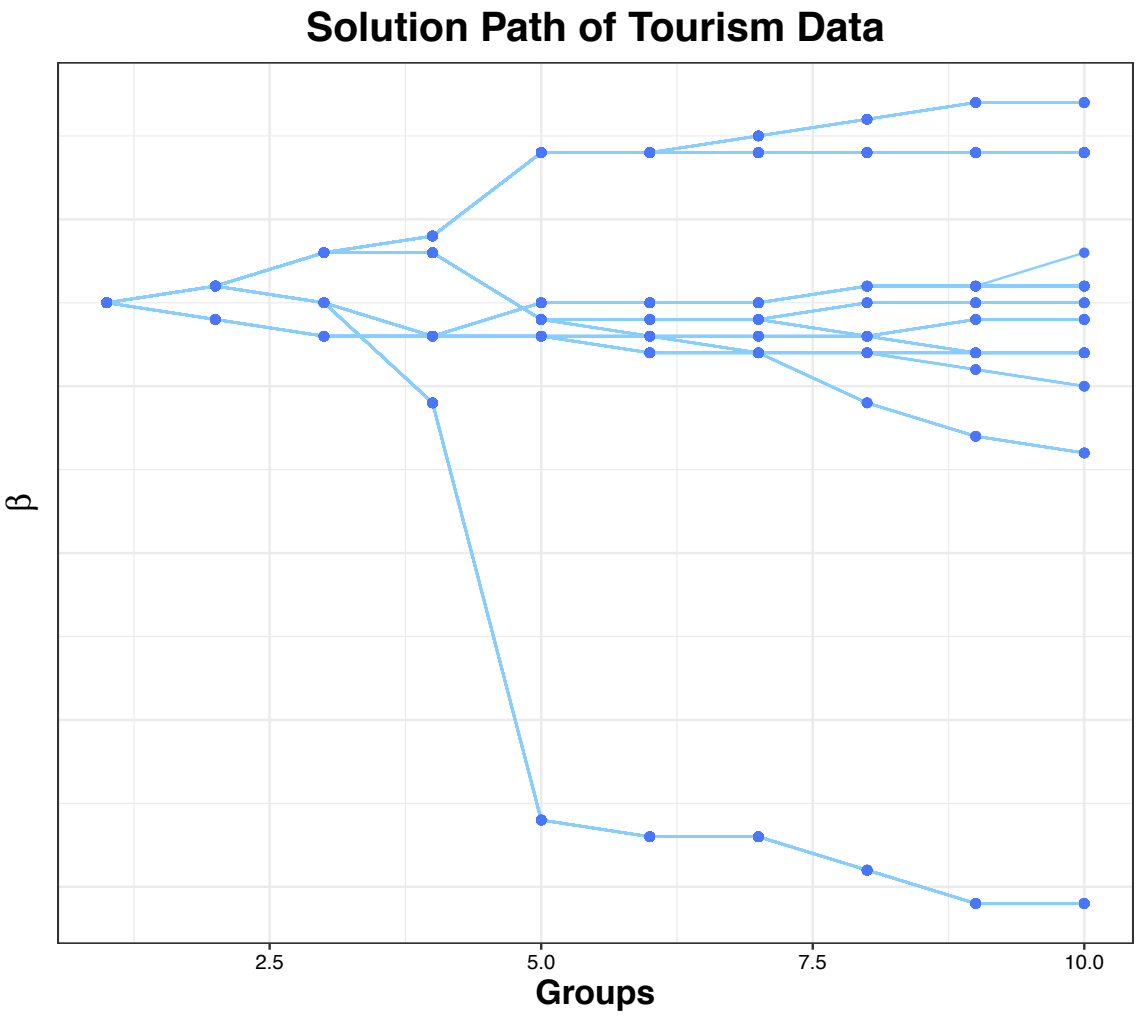


Figure 2.1: Example of a Solution Path.

The dilemma here is that we may never know the true grouping status beforehand. Thus, the basic idea is to get an initial estimation of $\boldsymbol{\beta}_i$ $(1 \leq i \leq n)$, and this pilot estimate is denoted by $\boldsymbol{B}_0 = (\boldsymbol{\beta}_{10}, \ldots, \boldsymbol{\beta}_{n0})$. Then we can use $\boldsymbol{B}_0$'s row-wise similarity to formulate the weight. Based on this idea, we propose two different weighting schemes:

1. Nearest Neighborhood Method: When λ is small and $w_{ij} = 1$ $(1 \leq i < j \leq n)$ in equation (3.6), the solution $\boldsymbol{B}$ would contain lots of distinct rows. It can naturally be used as a pilot estimation of $\boldsymbol{B}$. Furthermore, inspired by some past works (Chi and Lange (2013), Chen et al. (2015a)) on convex clustering, we use an empirical formula

$$w_{ij} = \begin{cases} 10e^{-\phi d_{ij}^2}, & \boldsymbol{\beta}_{0i}(\boldsymbol{\beta}_{0j}) \text{ is within the } k \text{ nearest neighbors of } \boldsymbol{\beta}_{0j}(\boldsymbol{\beta}_{0i}), \\ e^{-\phi d_{ij}^2}, & \text{otherwise} \end{cases} \tag{2.13}$$

where $d_{ij} = \|\boldsymbol{\beta}_{0i} - \boldsymbol{\beta}_{0j}\|_2$ $(1 \leq i < j \leq n)$. However, we realized that d_{ij} may contain some extremely large or small values in practice which could be too strong to be used in weight construction. Thus, we truncate extremely small d_{ij}s (smaller than its 10% quantile) to be exactly its 10% quantile, and extremely large d_{ij}s (larger than its 90% quantile) to its 90% quantile. Here, ϕ is a scale parameter to control the cohesive strength, and $\phi = 0$ corresponds to $w_{ij} = 1$. We set different thresholds (1 and 10) based on the neighborhood condition between $\boldsymbol{\beta}_{0i}$ and $\boldsymbol{\beta}_{0j}$. For tuning convenience, we scale the weight $\{w_{ij}\}_{1 \leq i < j \leq n}$ to the sum of 1 in the end.

2. Bayesian Method: Although the first method is convenient, it is not always stable. Since we get $\boldsymbol{B}_0$ from a plain weight which does not have very strong grouping power, the resulting structure of $\boldsymbol{B}_0$ might be misleading. As a result. We propose the second weighting scheme based on Bayesian inference to get $\boldsymbol{B}_0$. The whole process has three steps:

Step one: Fit a standard GLM using the whole data set. We generally recommend standardizing each predictor to ensure all variables are on the same scale. The estimated coefficients from the GLM is $\hat{\boldsymbol{\beta}}$ with standard deviation as $\hat{\sigma}_{\boldsymbol{\beta}}$.

Step two: For each sample $(\boldsymbol{x}_i, y_i)$ $(i = 1, \ldots, n)$, we can get an independent posterior estimation of $\boldsymbol{\beta}_i$ from GLM. The likelihood for a single observation is in equation (2.8). Bayesian analysis requires specifying prior distribution $f(\boldsymbol{\beta}_i)$. Based on the GLM from step one, it is natural to assume the prior distribution follow $\boldsymbol{\beta}_i \sim Normal(\hat{\boldsymbol{\beta}}, c\hat{\sigma}_{\boldsymbol{\beta}})$. c is positive constant, and we usually set it larger than 1 to allow more flexibility to its posterior estimation. The posterior distribution of $\boldsymbol{\beta}_i$ satisfies:

$$f(\boldsymbol{\beta}_i \mid y_i, \boldsymbol{x}_i) \propto f(\boldsymbol{\beta}_i) f(y_i \mid \boldsymbol{x}_i), \tag{2.14}$$

where $f(y_i \mid \boldsymbol{x}_i)$ is obtained from equation (2.8). We can draw this posterior distribution through Markov Chain Monte Carlo (MCMC) using Rstan (`https://mc-stan.org/docs/2_21/stan-users-guide/index.html`). We estimate $\boldsymbol{\beta}_i$ as its posterior mean. Since the estimation of intercepts is meaningless (because they are all estimated based on the same value 1), we drop it and denote the rest as $\hat{\boldsymbol{\beta}}_{i0}$. In this way, posterior estimations from each sample form a matrix $\boldsymbol{B}_0 = (\hat{\boldsymbol{\beta}}_{10}, \ldots, \hat{\boldsymbol{\beta}}_{n0})^T$. Intuitively, samples having similar posterior estimations should be more likely to belong to the same group.

Step three: Use $\boldsymbol{B}_0$ to generate w_{ij}s in two ways.

(2a). Principle Component Ranking In some cases, the direction with the most variation within $\boldsymbol{B}_0$ also contains critical group information. Thus, we can select the first principle component of $\boldsymbol{B}_0$ and use it to rank. Denote the first principle component as $\boldsymbol{z}_1 \in \mathbb{R}^n$. Then we can get the rank order of $\boldsymbol{z}_1$ as $\boldsymbol{r}_1 = (r_1 \ldots, r_n)^T \in \mathbb{R}^n$. The weight can be formulated as

$$w_{ij} = \frac{1}{|r_i - r_j|^k}, \quad 1 \le i < j \le n, \tag{2.15}$$

where $k > 0$ controls the cohesive strength. We can tune k or set a default value (like 2 or 3) in practice. We rescale $\{w_{ij}\}_{1 \le i < j \le n}$ to the sum of 1 in the end.

(2b). Relative Ranking. Assigning rank is easy for 1-dimensional arrays. But not for multi-dimensional arrays. Using the first principle component might lose some useful

information as well. Thus, we propose a ranking method that can be directly used on multi-dimensional arrays. Given $\boldsymbol{B}_0$, for any $\boldsymbol{\beta}_{i0}$ and $\boldsymbol{\beta}_{j0}$ $(1 \leq i < j \leq n)$, consider set $\mathcal{S}_1 = \{\|\boldsymbol{\beta}_{i0} - \boldsymbol{\beta}_{l0}\|_2 \mid k \neq i, 1 \leq k \leq n\}$ and set $\mathcal{S}_2 = \{\|\boldsymbol{\beta}_{j0} - \boldsymbol{\beta}_{l0}\|_2 \mid k \neq j, 1 \leq k \leq n\}$. Let r^*_{ij} be the rank order of $\|\boldsymbol{\beta}_{i0} - \boldsymbol{\beta}_{j0}\|_2$ in $\mathcal{S}_1$, r^*_{ji} be the rank order of $\|\boldsymbol{\beta}_{i0} - \boldsymbol{\beta}_{j0}\|_2$ in $\mathcal{S}_2$. Then we denote

$$r_{ij} = \frac{r^*_{ij} + r^*_{ji}}{2}$$

as the rank of $\|\boldsymbol{b}_{0i} - \boldsymbol{b}_{0j}\|_2$ among all pairwise differences related to $\boldsymbol{\beta}_{0i}$ and $\boldsymbol{\beta}_{0j}$. Let

$$w_{ij} = \frac{1}{r^k_{ij}}, \tag{2.16}$$

where $k > 0$ also controls the cohesive strength as in equation (2.15). We rescale $\{w_{ij}\}_{1 \leq i < j \leq n}$ to the sum of 1 in the end.

2.3 IOSA: Algorithm Formulation

It is easy to verify that $f(\boldsymbol{B} \mid \boldsymbol{y}, \boldsymbol{X})$ in equation (3.5) is convex, but solving the optimization problem is still nontrivial because of the non-linear objective function and many pairwise difference terms in the penalty. Our algorithm starts with a so-called "linearization" of $f(\boldsymbol{B} \mid \boldsymbol{y}, \boldsymbol{X})$. Then by solving a simpler optimization problem using operator splitting method at each step, we will get a set of iterates that converges to a global minimizer of the optimization problem (3.5). We also employ a momentum-based acceleration scheme (Beck and Teboulle 2009) to speed the convergence. The whole algorithm is names as IOSA (Iterative Operator-splitting for Samples Algorithm) and is summarized as below:

Algorithm 1 IOSA

Initialize $\boldsymbol{B}^{[0]} = \boldsymbol{B}^{[1]}$, $\theta_0 = 1$.

for $k = 1, 2, 3, \ldots$ **do**

$\theta^{[k]} = (\sqrt{(\theta^{[k-1]})^4 + 4(\theta^{[k]})^2} - (\theta^{[k]})^2)/2$

$\boldsymbol{B}^{[k]}_{acc} = \boldsymbol{B}^{[k]} + \theta^{[k]}(1/\theta^{[k-1]} - 1)(\boldsymbol{B}^{[k]} - \boldsymbol{B}^{[k-1]})$

$\boldsymbol{Z}^{[k]} = \boldsymbol{B}^{[k]}_{acc} - \psi(\boldsymbol{B}^{[k]}_{acc})/\rho$ where $\psi(\cdot)$ is defined in equation (2.18)

Solve the inner optimization problem:

$$\boldsymbol{B}^{[k+1]} = \arg\min_{\boldsymbol{B}} \frac{1}{2}\|\boldsymbol{Z}^{[k]} - \boldsymbol{B}\|_F^2 + \tilde{\lambda} w_{ij} \sum_{1 \le i < j \le n} \|\boldsymbol{\beta}_i - \boldsymbol{\beta}_j\|_2,$$

where $\tilde{\lambda}$ is a known parameter to be specified later.

We use the following relative error as the stopping criteria.

$$\frac{\|\boldsymbol{B}^{[k+1]} - \boldsymbol{B}^{[k]}\|_\infty}{\|\boldsymbol{B}^{[k]}\|_\infty} < \epsilon,$$

where $\epsilon > 0$ is a small positive value, say $1e-3$. Alternatively, we can also calculate the duality gap as the stopping criteria. A derivation of the duality gap is given in Appendix A.2.

In the next section, we will show the details of the algorithm design

2.3.1 Linearization

First, we introduce a surrogate function. Denote $\boldsymbol{X}_i$ to be a matrix of zeros except for the i-th row to be $\boldsymbol{x}_i^T$. Given any matrix $\boldsymbol{B}^- \in \mathbb{R}^{n \times p}$, the surrogate function of $\boldsymbol{B}$ at $\boldsymbol{B}^-$ is defined as

$$\begin{aligned} G(\boldsymbol{B}, \boldsymbol{B}^-) =& l(\boldsymbol{B}^-) + \Big\langle \sum_{i=1}^{n} \{\tau'(\langle \boldsymbol{X}_i, \boldsymbol{B}^- \rangle_F) - y_i\} \boldsymbol{X}_i, \boldsymbol{B} - \boldsymbol{B}^- \Big\rangle_F \\ &+ \frac{\rho}{2}\|\boldsymbol{B} - \boldsymbol{B}^-\|_F^2 + P(\boldsymbol{B}). \end{aligned} \tag{2.17}$$

Here $\rho > 0$ is the inverse step size which can be chosen later. Denote

$$\psi(\boldsymbol{B}) := \sum_{i=1}^{n} \{\tau'(\langle \boldsymbol{X}_i, \boldsymbol{B}\rangle_F) - y_i\}\boldsymbol{X}_i$$

. We have the matrix representation,

$$\psi(\boldsymbol{B}) = \mathrm{diag}\{[\tau'(\boldsymbol{\eta}) - \boldsymbol{y}]\}\boldsymbol{X}.$$

Now, given any initial point $\boldsymbol{B}^{[0]} \in \mathbb{R}^{n\times p}$, a sequential of $\boldsymbol{B}$ is iteratively determined by

$$\boldsymbol{B}^{[k+1]} = \arg\min_{\boldsymbol{B}} G(\boldsymbol{B}, \boldsymbol{B}^{[k]}), \quad k = 0, 1, \ldots.$$

Theorem 2.3.1 shows that $\boldsymbol{B}^{[k]}$ will converge to the solution of the minimization problem (3.5). The proof of the theorem can be found in Appendix A.1.

Theorem 2.3.1. *Let* $\boldsymbol{A} = \sum_{i=1}^{n} vec(\boldsymbol{X}_i)vec(\boldsymbol{X}_i)^T$. *As long as* $\rho \geq \|\boldsymbol{A}\|_2/4$, *for all* $k \geq 0$,

$$f(\boldsymbol{B}^{[k+1]}) \leq G(\boldsymbol{B}^{[k+1]}, \boldsymbol{B}^{[k]}) \leq G(\boldsymbol{B}^{[k]}, \boldsymbol{B}^{[k]}) \leq f(\boldsymbol{B}^{[k]}).$$

i.e., the function values are non-increasing, and thus are convergent. Furthermore, when $\rho > \|\boldsymbol{A}\|_2^2/4$, *the sequence of iterates* $\boldsymbol{B}^{[k]}$ *converges.*

Remark 1. The inequalities might be strengthened to prove the convergence of iterates. For example:

$$G(\boldsymbol{B}^{[k]}, \boldsymbol{B}^{[k]}) - G(\boldsymbol{B}^{[k+1]}, \boldsymbol{B}^{[k]}) \geq \frac{\rho}{2}\|\boldsymbol{B}^{[k+1]} - \boldsymbol{B}^{[k]}\|_F^2.$$

Remark 2. In our scenario, we actually have

$$\|\boldsymbol{A}\|_2 = \|\boldsymbol{X}\|_{2,\infty}^2 = \max_{1\leq i\leq n} \|\boldsymbol{x}_i\|_2^2.$$

After linearization, the problem boils down to solving

$$\min_{\boldsymbol{B}\in\mathbb{R}^{n\times p}} \frac{1}{2}\left\|\boldsymbol{B}^{[k]} - \frac{\sum_{i=1}^{n}\{\tau'(\langle \boldsymbol{X}_i, \boldsymbol{B}^{[k]}\rangle_F) - y_i\}\boldsymbol{X}_i}{\rho} - \boldsymbol{B}\right\|_F^2 + \frac{1}{\rho}P(\boldsymbol{B}).$$

Let $\boldsymbol{Z} = \boldsymbol{B}^{[k]} - \psi(\boldsymbol{B}^{[k]})/\rho$ and $\tilde{\lambda} = \frac{\lambda}{\rho}$. We can solve the optimization problem (3.5) by iteratively solving the following quadratic optimization problem:

$$\min_{\boldsymbol{B}} \frac{1}{2}\|\boldsymbol{Z} - \boldsymbol{B}\|_F^2 + \tilde{\lambda}\sum_{1\le i<j\le n} w_{i,j}\|\boldsymbol{\beta}_i - \boldsymbol{\beta}_j\|_2. \tag{2.18}$$

Remark 3. The previous algorithm is a first-order one, using only the gradient information to update the iterates. A Newton or Quasi-Newton algorithm can be derived instead, such as *minFunc: unconstrained differentiable multivariate optimization in Matlab.* (Appendix A.3 provides a derivation based on it). In comparison, our algorithm is more scalable in higher dimensions, although in lower dimensions it may converge slower.

In the next section, we will introduce three different algorithms to solve the inner optimization problem in equation (2.18).

2.3.2 Operator Splitting for the g-optimizer

We introduce three different algorithms to solve (2.18). We first define some matrices to prepare for the algorithm derivation.

We introduce a pairwise difference matrix $\boldsymbol{T} = (\boldsymbol{t}_1, \ldots, \boldsymbol{t}_l) = \{t_{i,j}\} \in \mathbb{R}^{l\times n}$ where $l = \frac{n(n-1)}{2}$

$$\boldsymbol{T} := \begin{pmatrix} 1 & -1 & 0 & \ldots & 0 \\ 1 & 0 & -1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 1 & -1 \end{pmatrix}_{l\times n}. \tag{2.19}$$

$\boldsymbol{T}$ does not have full rank and $rank(\boldsymbol{T}) = n-1$. We introduce an index one-on-one mapping

$\mathcal{F}$ from the set of indexes $\mathcal{S}_3 := \{1 \ldots, l\}$ to $\mathcal{S}_4 = \{(i,j) \mid 1 \le i < j \le n\}$:

$$\mathcal{F}(i) = (i_1, i_1), \quad s.t.\ t_{i,i_1} = 1,\ t_{i,i_2} = -1.$$

In this way, the weight value w_{ij}s $(1 \le i < j \le n)$ could be expressed as a vector of length l. Namely

$$\boldsymbol{w} = (w_{\mathcal{F}(1)}, \ldots, w_{\mathcal{F}(l)})^T \in \mathbb{R}^l. \tag{2.20}$$

Then we can define diagonal matrices $\boldsymbol{\Lambda} \in \mathbb{R}^{l \times l}$ and $\tilde{\boldsymbol{\Lambda}} \in \mathbb{R}^{\tilde{l} \times \tilde{l}}$ where $\tilde{l} := l + 1$

$$\boldsymbol{\Lambda} = \tilde{\lambda} \cdot \mathrm{diag}(\boldsymbol{w}), \quad \tilde{\boldsymbol{\Lambda}} = \tilde{\lambda} \cdot \mathrm{diag}(0, \boldsymbol{w}). \tag{2.21}$$

We obtain the SVD of $\boldsymbol{T}$ from She (2010) and let $\boldsymbol{T} = \boldsymbol{U}_0 \boldsymbol{D}_0 \boldsymbol{V}_0^T$, where $\boldsymbol{U}_0$, $\boldsymbol{D}_0$, $\boldsymbol{V}_0$ are

$$\boldsymbol{U}_0 = \left(u_{21} \quad \tfrac{1}{\sqrt{n}} \boldsymbol{T} \boldsymbol{V}_1\right)_{l \times n}, \quad u_{21} = \frac{1}{\sqrt{3}} \begin{pmatrix} 0 & \cdots & 0 & 1 & -1 & 1 \end{pmatrix}^T_{l \times 1},$$

$$\boldsymbol{V}_1 = \sqrt{\frac{2}{n}} \left(\cos\left(\tfrac{(2i-1)j\pi}{2n}\right)\right)_{n \times (n-1)}, \quad \boldsymbol{D}_0 = \mathrm{diag}\begin{pmatrix} 0 & \sqrt{n} & \cdots & \sqrt{n} \end{pmatrix}_{n \times n},$$

$$\boldsymbol{V}_0 = \left(\tfrac{1}{\sqrt{n}} 1_{n \times 1} \quad \boldsymbol{V}_1\right)_{n \times n}.$$

Here $\boldsymbol{U}_0$ and $\boldsymbol{V}_0$ satisfy

$$\boldsymbol{U}_0^T \boldsymbol{U}_0 = \boldsymbol{I}, \quad \boldsymbol{V}_0^T \boldsymbol{V}_0 = \boldsymbol{V}_0 \boldsymbol{V}_0^T = \boldsymbol{I}.$$

Sometimes, we want to add a row of 1s to $\boldsymbol{T}$ to get a full column rank $\tilde{\boldsymbol{T}} \in \mathbb{R}^{\tilde{l}\times n}$

$$\tilde{T} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 0 & \dots & 0 \\ 1 & 0 & -1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 1 & -1 \end{pmatrix}_{\tilde{l}\times n} . \tag{2.22}$$

Denote the SVD of $\tilde{\boldsymbol{T}}$ as $\tilde{\boldsymbol{T}} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T$. Let

$$\boldsymbol{H} := \boldsymbol{V}\boldsymbol{D}^{-1}\boldsymbol{U}^T, \quad \tilde{\boldsymbol{C}} := \tilde{\boldsymbol{T}}\boldsymbol{B}. \tag{2.23}$$

We have

$$\boldsymbol{H}\tilde{\boldsymbol{T}} = \boldsymbol{I}, \quad \boldsymbol{B} = \boldsymbol{H}\tilde{\boldsymbol{C}}. \tag{2.24}$$

The SVD of $\boldsymbol{T}$ is

$$\boldsymbol{U} = \begin{pmatrix} 1 & 1 \\ 0 & \frac{1}{\sqrt{n}}\boldsymbol{T}\boldsymbol{V}_1 \end{pmatrix}_{\tilde{l}\times n}$$

and

$$\boldsymbol{D} = \mathrm{diag}(\sqrt{n}, \dots, \sqrt{n})_{n\times n}, \quad \boldsymbol{V} = \boldsymbol{V}_0$$

satisfying

$$\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}, \quad \boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{V}\boldsymbol{V}^T = \boldsymbol{I}. \tag{2.25}$$

We will use these matrix notation to introduce three algorithms to solve equation (2.18).

2.3.3 Algorithm One: ADMM

ADMM (Alternating Direction Method of Multipliers) (Boyd, Parikh, and Chu 2011) is a popular algorithm to solve convex optimization problems. Its basic idea is to break the

problem into smaller pieces by introducing an ancillary operator, denote as $\boldsymbol{C} := \boldsymbol{TB} \in \mathbb{R}^{l\times p}$. Since the linearized problem in equation (2.18) is convex but not strongly convex, we can add an augmented term to make it strongly convex and get an equivalent problem

$$\min_{\boldsymbol{B},\boldsymbol{C}} \frac{1}{2}\|\boldsymbol{Z}-\boldsymbol{B}\|_F^2 + \|\boldsymbol{\Lambda C}\|_{2,1} + \frac{\nu}{2}\|\boldsymbol{TB}-\boldsymbol{C}\|_F^2, \quad s.t.\ \boldsymbol{TB}=\boldsymbol{C},$$

where the $l_{2,1}$-norm, $\boldsymbol{\Lambda}$, $\boldsymbol{T}$ are defined in equation (2.4), (2.19), (2.21), and $\nu > 0$. The Lagrangian is

$$L_\nu(\boldsymbol{B},\boldsymbol{C},\boldsymbol{\Gamma}) = \frac{1}{2}\|\boldsymbol{Z}-\boldsymbol{B}\|_F^2 + \|\boldsymbol{\Lambda C}\|_{2,1} + \frac{\nu}{2}\|\boldsymbol{TB}-\boldsymbol{C}\|_F^2 + \langle \boldsymbol{\Gamma}, \boldsymbol{TB}-\boldsymbol{C}\rangle_F,$$

where $\boldsymbol{\Gamma} = (\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_l)^T \in \mathbb{R}^{l\times p}$ is the Lagrangian multiplier. Since jointly minimizing $\boldsymbol{B}$ and $\boldsymbol{C}$ is difficult, we minimize $\boldsymbol{B}$ and $\boldsymbol{C}$ separately. This yields

$$\boldsymbol{B}^{[k+1]} = \arg\min_{\boldsymbol{B}} L_\nu(\boldsymbol{B},\boldsymbol{C}^{[k]},\boldsymbol{\Gamma}^{[k]}), \tag{2.26}$$

$$\boldsymbol{C}^{[k+1]} = \arg\min_{\boldsymbol{C}} L_\nu(\boldsymbol{B}^{[k+1]},\boldsymbol{C},\boldsymbol{\Gamma}^{[k]}), \tag{2.27}$$

$$\boldsymbol{\Gamma}^{[k+1]} = \boldsymbol{\Gamma}^{[k]} + \nu(\boldsymbol{TB}^{[k+1]} - \boldsymbol{C}^{[k+1]}). \tag{2.28}$$

To solve equation (2.26), we can take the partial derivative of $L_\nu(\cdot,\boldsymbol{C},\boldsymbol{\Gamma})$ with respect to $\boldsymbol{B}$ and set it to 0. We have

$$(\boldsymbol{I}+\nu\boldsymbol{T}^T\boldsymbol{T})\boldsymbol{B} = \boldsymbol{Z} + \nu\boldsymbol{T}^T(\boldsymbol{C}-\frac{1}{\nu}\boldsymbol{\Gamma}^T).$$

It's easy to verify that $\boldsymbol{TT}^T = n\boldsymbol{I} - n11^T$, using Sherman-Morrison formula (Chavez 2006), we can get $(\boldsymbol{I}+\nu\boldsymbol{TT}^T)^{-1} = \frac{1}{1+n\nu}[\boldsymbol{I}+\nu 11^T]$. Thus,

$$\boldsymbol{B} = \frac{1}{1+n\nu}(\boldsymbol{I}+\nu 11^T)[\boldsymbol{Z}+\nu\boldsymbol{T}^T(\boldsymbol{C}-\frac{1}{\nu}\boldsymbol{T}^T)].$$

On the other hand, Solving equation (2.27) is equivalent to solving

$$\arg\min_{C} \frac{1}{2}[\|C - (\boldsymbol{TB} - \frac{1}{\nu}\boldsymbol{\Gamma})\|_F^2 + \frac{1}{\nu}\|\boldsymbol{\Lambda C}\|_{2,1}].$$

The above equation for $\boldsymbol{C} = (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_l)^T$ is separable for each row. Thus we can get $\boldsymbol{c}_i$ by the proximal mapping and get

$$\boldsymbol{c}_i = \mathbf{prox}_{\sigma\|\cdot\|_2}(\boldsymbol{t}_i\boldsymbol{B} - \frac{1}{\nu}\boldsymbol{\gamma}_i), \quad i = 1, \ldots, l,$$

where $\sigma_l = \tilde{\lambda} w_l / \nu$.

In the summary, we can get the following algorithm:

Algorithm 2 ADMM

Initialize $\boldsymbol{\Gamma}^{[0]}$ and $\boldsymbol{C}^{[0]}$

for $k = 1, 2, 3, \ldots$ **do**

for $j = 1, \ldots, n$ **do**

$\boldsymbol{p}_j = \boldsymbol{z}_j + \sum_{i:i_1=j}(\nu \boldsymbol{c}_i^{[k-1]} + \boldsymbol{\gamma}_i^{[k-1]}) - \sum_{i:i_2=j}(\nu \boldsymbol{c}_i^{[k-1]} + \boldsymbol{\gamma}_i^{[k-1]})$

$\boldsymbol{B}^{[k]} = \frac{1}{1+n\nu}\boldsymbol{P}^T + \frac{n\nu}{1+n\nu}\bar{\boldsymbol{Z}}, \quad \boldsymbol{P} = (\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n)^T, \quad \bar{\boldsymbol{Z}} = \frac{1}{n}\boldsymbol{Z}^T 1_n 1_n^T$

for $i = 1, \ldots, l$ **do**

$\boldsymbol{c}_i^{[k]} = prox_{\sigma_l}(\boldsymbol{\beta}_{i1}^{[k]} - \boldsymbol{\beta}_{i2}^{[k]} - \nu^{-1}\boldsymbol{\gamma}_i^{[k-1]})$, $\sigma_i = \tilde{\lambda} w_i / \nu$

$\boldsymbol{\Gamma}_l^{[k]} = \boldsymbol{\Gamma}_l^{[k-1]} + \nu(\boldsymbol{c}_l^{[k]} - \boldsymbol{\beta}_{i1}^{[k]} + \boldsymbol{\beta}_{i2}^{[k]})$

The stopping criteria is formed based on the primal and dual residuals given by Boyd as below:

$$\begin{cases} \boldsymbol{p}_i^{[k+1]} = \boldsymbol{\beta}_{i1}^{[k+1]} - \boldsymbol{\beta}_{I_{is}}^{[k+1]} - \boldsymbol{c}_i^{[k+1]}, \quad i = 1, \ldots, n, \\ \boldsymbol{d}_i^{[k+1]} = -\nu(\sum_{j_1=i}(\boldsymbol{c}_l^{[k+1]} - \boldsymbol{c}_l^{[k]}) - \sum_{j_2=i}(\boldsymbol{c}_l^{[k+1]} - \boldsymbol{c}_l^{[k]})), \end{cases} \tag{2.29}$$

We stop the algorithm until $\|\boldsymbol{p}_i^{[k+1]} - \boldsymbol{d}_i^{[k+1]}\|_2$ is small enough.

2.3.4 Algorithm Two: AMA

AMA (Alternating Minimization Algorithm) is proposed by Tseng (1991) to solve convex optimization problems with two-block separable linear constraints and objectives. Compared with ADMM which augments $\boldsymbol{B}$ and $\boldsymbol{C}$ together, it only augments the term which is not strongly convex. It borrows strength from proximal gradient method (Parikh and Boyd 2014) and gets rid of the complete augmentation in ADMM.

The linearized problem in equation (2.18) is equivalent to

$$\min_{\boldsymbol{B},\boldsymbol{C}} \frac{1}{2}\|\boldsymbol{Z}-\boldsymbol{B}\|_F^2 + \|\boldsymbol{\Lambda C}\|_{2,1}, \quad s.t.\ \boldsymbol{TB}=\boldsymbol{C}. \tag{2.30}$$

The Lagrangian is

$$L(\boldsymbol{B},\boldsymbol{C},\boldsymbol{\Gamma}) = \frac{1}{2}\|\boldsymbol{Z}-\boldsymbol{B}\|_F^2 + \|\boldsymbol{\Lambda C}\|_{2,1} + \langle \boldsymbol{\Gamma}, \boldsymbol{TB}-\boldsymbol{C}\rangle_F, \tag{2.31}$$

where $\boldsymbol{\Gamma} = (\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_l)^T \in \mathbb{R}^{l\times p}$ is the Lagrangian multiplier. To minimize the Lagrangian w.r.t. $\boldsymbol{B}$ and $\boldsymbol{C}$, we first take the partial derivative of $L(\cdot,\boldsymbol{C},\boldsymbol{\Gamma})$ with respect to $\boldsymbol{B}$ and get

$$\boldsymbol{B} = \boldsymbol{Z} - \boldsymbol{T}^T\boldsymbol{\Gamma}.$$

Substituting it into equation (2.31), we get the dual problem

$$\max_{\boldsymbol{\Gamma}} D(\boldsymbol{\Gamma}) = \frac{1}{2}\|\boldsymbol{Z}-(\boldsymbol{Z}-\boldsymbol{T}^T\boldsymbol{\Gamma})\|_F^2 - (\|\boldsymbol{\Lambda}\cdot\|_{2,1})^*(\boldsymbol{\Gamma}) + \langle \boldsymbol{\Gamma}, \boldsymbol{T}(\boldsymbol{T}^T\boldsymbol{\Gamma}-\boldsymbol{Z})\rangle_F,$$

where $(\|\boldsymbol{\Lambda}\cdot\|_{2,1})^*(\boldsymbol{\Gamma}) = \sup_{\boldsymbol{C}}\langle \boldsymbol{C},\boldsymbol{\Gamma}\rangle_F - \|\boldsymbol{\Lambda C}\|_{2,1}$ is the Fenchel conjugate of $\|\boldsymbol{\Lambda}\cdot\|_{2,1}$. Since the problem is convex with equality constraint, strong duality holds true. The dual problem

is equivalent to

$$\begin{aligned}\max_{\boldsymbol{\Gamma}} D(\boldsymbol{\Gamma}) &= \frac{1}{2}\|\boldsymbol{Z}\|_F^2 - \|\boldsymbol{T}^T\boldsymbol{\Gamma} - \boldsymbol{Z}\|_F^2/2 - (\|\boldsymbol{\Lambda}\cdot\|_{2,1})^*(\boldsymbol{\Gamma}) \\ \Leftrightarrow \min_{\boldsymbol{\Gamma}} &\|\boldsymbol{T}^T\boldsymbol{\Gamma} - \boldsymbol{Z}\|_F^2/2 + (\|\boldsymbol{\Lambda}\cdot\|_{2,1})^*(\boldsymbol{\Gamma}). \end{aligned} \tag{2.32}$$

We can use proximal gradient method to solve equation (2.32). Let $f(\boldsymbol{\Gamma}) := \|\boldsymbol{T}^T\boldsymbol{\Gamma} - \boldsymbol{Z}\|_F^2/2$. Assume ∇f is Lipschitz continuous with constant L, then proximal gradient update will converge with first order rate when $v \in (0, \frac{1}{L}]$. As a result, we can get the following update of $\boldsymbol{\Gamma}$:

$$\boldsymbol{\Gamma}^{[k+1]} = prox_{v\|\boldsymbol{\Lambda}\cdot\|_{2,1}^*}(\boldsymbol{\Gamma}^{[k]} - v(\boldsymbol{T}\boldsymbol{T}^T\boldsymbol{\Gamma}^{[k]} - \boldsymbol{T}\boldsymbol{Z})), \quad v \in (0, \frac{1}{L}], \quad k = 0, 1, \ldots.$$

Chi and Lange (2013) pointed out that empirically $v < \frac{2}{n}$ often works when there are fewer than 1000 data points. The Fenchel conjugate of $\|\tilde{\boldsymbol{\Lambda}}\cdot\|_{2,1}$ can be obtained row-wisely. It is easy to prove that for any vector $\boldsymbol{z}$, function $h(\boldsymbol{x}) := \|\boldsymbol{x}\|_2$'s Fenchel conjugate is $h^*(\boldsymbol{x}) = \delta_B(\boldsymbol{x})$. i.e., the delta function within the unit ball $\mathcal{B} = \{\boldsymbol{x} : \|\boldsymbol{x}\|_2^* \leq 1\}$ where $\|\cdot\|_2^*$ is the dual norm. Since the proximal mapping of the delta function of a closed convex set is equivalent to projection ($\mathcal{P}(\cdot)$) onto the set, we finally get the Fenchel conjugate of $\|\boldsymbol{\Lambda}\cdot\|_{2,1}$ to be

$$prox_{v\|\boldsymbol{\Lambda}\cdot\|_{2,1}^*}(\boldsymbol{\Gamma}) = (prox_{v\delta_{C_i}}(\boldsymbol{\gamma}_i))_{i=1}^l = (\mathcal{P}_{\delta_{C_i}}(\boldsymbol{\gamma}_i))_{i=1}^l$$

where $\mathcal{B}_i = \{\boldsymbol{\gamma}_i : \|\boldsymbol{\gamma}_i\|^* \leq \tilde{\lambda} w_i\}$. The identity $v\delta_{\mathcal{B}_i} = \delta_{\mathcal{B}_i}$ holds because $\delta_{\mathcal{B}_i}$ only takes value 0 and ∞.

Note that we actually do not need to update $\boldsymbol{C}$ explicitly, and it's easy to verify that updating $\boldsymbol{C}$ is equivalent to minimizing the augmented Lagrangian

$$min_{\boldsymbol{C}}\frac{1}{2}\|\boldsymbol{Z} - \boldsymbol{B}\|_F^2 + \|\boldsymbol{\Lambda}\boldsymbol{C}\|_{2,1} + \langle\boldsymbol{\Gamma}, \boldsymbol{T}\boldsymbol{B} - \boldsymbol{C}\rangle_F + \frac{v}{2}\|\boldsymbol{T}\boldsymbol{B} - \boldsymbol{C}\|_F^2.$$

In the summary, we can get the following algorithm:

Algorithm 3 AMA

procedure

Initialize $\mathbf{\Gamma}^{[0]}$

for $k = 1, 2, 3, \ldots$ **do**

for $j = 1, \ldots, n$ **do**

$\Delta_j^{[k]} = \sum_{i:i_1=j} \boldsymbol{\gamma}_i^{[k-1]} - \sum_{i:i_2=j} \boldsymbol{\gamma}_i^{[k-1]}$.

for $i = 1, \ldots, l$ **do**

$\boldsymbol{h}_i^{[k]} = \boldsymbol{z}_{i_1} - \boldsymbol{z}_{i_2} + \Delta_{i_1}^{[k]} - \Delta_{i_2}^{[k]}$

$\boldsymbol{\gamma}_i^{[k]} = \mathcal{P}_{\mathcal{B}_i}(\boldsymbol{\gamma}_i^{[k-1]} - \nu \boldsymbol{h}_i^{[k]})$, where $\mathcal{B}_i = \{\boldsymbol{\gamma}_i : \|\boldsymbol{\gamma}_i\|_2 \leq \tilde{\lambda} w_i\}$

Furthermore, this algorithm can be accelerated through Nesterov's acceleration (Beck and Teboulle 2009). The accelerated version is as bellow:

Algorithm 4 Accelerated AMA

Initialize $\mathbf{\Gamma}^{[-1]}$, $\alpha^{[0]} = 1$

for $k = 0, 1, 2, \ldots$ **do**

for $j = 1, \ldots, n$ **do**

$\Delta_j^{[k]} = \sum_{i:i_1=j} \boldsymbol{\gamma}_i^{[k-1]} - \sum_{i:i_2=j} \boldsymbol{\gamma}_i^{[k-1]}$.

for $i = 1, \ldots, l$ **do**

$\boldsymbol{h}_i^{[k]} = \boldsymbol{z}_{i_1} - \boldsymbol{z}_{i_2} + \Delta_{i_1}^{[k]} - \Delta_{i_2}^{[k]}$

$\tilde{\boldsymbol{\gamma}}_i^{[k]} = \mathcal{P}_{\mathcal{B}_i}(\boldsymbol{\gamma}_i^{[k-1]} - \nu \boldsymbol{h}_i^{[k]})$, where $\mathcal{B}_i = \{\boldsymbol{\gamma}_i : \|\boldsymbol{\gamma}_i\|_2 \leq \tilde{\lambda} w_i\}$.

Let $\alpha^{[k]} := (1 + \sqrt{1 + 4(\alpha^{[k-1]})^2})/2$

$\boldsymbol{\gamma}_i^{[k+1]} = \tilde{\boldsymbol{\gamma}}_i^k + \frac{\alpha^{[k-1]}}{\alpha^{[k]}}[\tilde{\boldsymbol{\gamma}}_i^{[k]} - \tilde{\boldsymbol{\gamma}}_i^{[k-1]}]$

We can use duality gap as the stopping criteria. Define

$$P(\boldsymbol{B}^{[k]}) = \frac{1}{2}\|\boldsymbol{Z} - \boldsymbol{B}^{[k]}\|_F^2 + \|\boldsymbol{\Lambda}\boldsymbol{T}\boldsymbol{B}^{[k]}\|_{2,1},$$
$$D(\boldsymbol{\Gamma}^{[k]}) = -\frac{1}{2}\|\boldsymbol{T}^T\boldsymbol{\Gamma}^{[k]}\|_F^2/2 - \sum_{i=1}^{l} \iota_{\|\boldsymbol{\gamma}_i\|_2 \le \tilde{\lambda} w_i} - \langle \boldsymbol{\Gamma}^{[k]}, \boldsymbol{T}\boldsymbol{Z}\rangle_F.$$

We will stop if $|P(\boldsymbol{B}^{[k]}) - D(\boldsymbol{\Gamma}^{[k]})| \le \epsilon$ with a pre-specified small $\epsilon > 0$.

2.3.5 Algorithm Three: Dykstra Projection

The previous two algorithms are both based $\boldsymbol{B}$ and $\boldsymbol{C}$. For the third algorithm, we introduce the Dykstra's prpjection algorithm (Combettes and Pesquet 2011). We re-formulate equation (2.18) in terms of $\tilde{\boldsymbol{C}}$ defined in equation (2.23). Since $\tilde{\boldsymbol{T}}$ in equation (2.22) is complemented from $\boldsymbol{T}$, we call this algorithm as complemented minimization algorithm.

Similar as before, using $\boldsymbol{H}$ defined in equation (2.23) and combining equation (2.24), we can define a surrogate function of $\tilde{\boldsymbol{C}}$ at $\tilde{\boldsymbol{C}}^*$

$$\begin{aligned} G(\tilde{\boldsymbol{C}}, \tilde{\boldsymbol{C}}^*) =& l(\boldsymbol{C}^*) + \langle \sum_{i=1}^{n} \{\tau'(\langle \boldsymbol{H}^T\boldsymbol{X}_i, \tilde{\boldsymbol{C}}^*\rangle_F) - y_i\}\boldsymbol{H}^T\boldsymbol{X}_i, \tilde{\boldsymbol{C}} - \tilde{\boldsymbol{C}}^*\rangle_F \\ &+ \frac{\rho}{2}\|\tilde{\boldsymbol{C}} - \tilde{\boldsymbol{C}}^*\|_F^2 + P(\tilde{\boldsymbol{C}}), \end{aligned}$$

where $P(\tilde{\boldsymbol{C}}) := \|\tilde{\boldsymbol{\Lambda}}\tilde{\boldsymbol{C}}\|_{2,1}$. Denote

$$\begin{aligned} \psi(\tilde{\boldsymbol{C}}) &= \sum_{i=1}^{n} \{\tau'(\langle \boldsymbol{H}^T\boldsymbol{X}_i, \tilde{\boldsymbol{C}}^*\rangle_F) - y_i\}\boldsymbol{H}^T\boldsymbol{X}_i \\ &= \boldsymbol{H}^T \mathrm{diag}\{[\tau'(\boldsymbol{\eta}) - \boldsymbol{y}]\}\boldsymbol{X} \end{aligned}$$

As a result, given $\tilde{\boldsymbol{C}}^{[k]}$ at step k, we can update $\tilde{\boldsymbol{C}}^{[k+1]}$ through

$$\tilde{\boldsymbol{C}}^{[k+1]} = \arg\min_{\tilde{\boldsymbol{C}}} G(\tilde{\boldsymbol{C}}, \tilde{\boldsymbol{C}}^{[k]}).$$

Let $\tilde{\boldsymbol{A}} = \sum_{i=1}^{n} vec(\boldsymbol{H}^T\boldsymbol{X}_i)vec(\boldsymbol{H}^T\boldsymbol{X}_i)^T$. Using the similar proof in Theorem 2.3.1, we can get that if $\rho \geq \frac{\|\tilde{\boldsymbol{H}}\|_2}{4}$, we have:

$$f(\tilde{\boldsymbol{C}}^{[k+1]}) \leq G(\tilde{\boldsymbol{C}}^{[k+1]}, \tilde{\boldsymbol{C}}^{[k]}) \leq G(\tilde{\boldsymbol{C}}^{[k+1]}, \tilde{\boldsymbol{C}}^{[k+1]}) \leq f(\tilde{\boldsymbol{C}}^{[k]}).$$

As a result, the problem boils down to solving

$$\min_{\tilde{\boldsymbol{C}}} \frac{1}{2}\|\tilde{\boldsymbol{C}}^{[k]} - \frac{\sum_{i=1}^{n}\{\tau'(\langle \boldsymbol{H}^T\boldsymbol{X}_i, \tilde{\boldsymbol{C}}^{[k]}\rangle) - y_i\}\boldsymbol{H}^T\boldsymbol{X}_i}{\rho} - \tilde{\boldsymbol{C}}\|_F^2 + \frac{1}{\rho}P(\tilde{\boldsymbol{C}})$$

Let $\tilde{Z} = \tilde{\boldsymbol{C}}^{[k]} - \psi(\tilde{\boldsymbol{C}}^{[k]})/\rho$, $\tilde{\lambda} = \frac{\lambda}{\rho}$. From equation (2.24) and (2.25), we have

$$\boldsymbol{T}\boldsymbol{H}\tilde{\boldsymbol{C}} = \tilde{\boldsymbol{C}} \Leftrightarrow \boldsymbol{U}\boldsymbol{U}^T\tilde{\boldsymbol{C}} = \tilde{\boldsymbol{C}} \Leftrightarrow \boldsymbol{U}_\perp\boldsymbol{U}_\perp^T\tilde{\boldsymbol{C}} = 0 \Leftrightarrow \boldsymbol{U}_\perp^T\tilde{\boldsymbol{C}} = 0 \Leftrightarrow \tilde{\boldsymbol{C}} = \boldsymbol{U}\Phi, \quad \forall\Phi \in \mathbb{R}^{n\times p},$$

where $\boldsymbol{U}_\perp$ is the orthogonal complement of $\boldsymbol{U}$. The linearized problem is

$$\min_{\tilde{\boldsymbol{C}}} \frac{1}{2}\|\tilde{\boldsymbol{Z}} - \tilde{\boldsymbol{C}}\|_F^2 + \|\tilde{\boldsymbol{\Lambda}}\boldsymbol{C}\|_{2,1}, \quad s.t. \quad \tilde{\boldsymbol{C}} = \boldsymbol{U}\Phi, \quad \forall\Phi \in \mathbb{R}^{n\times p}.$$

It is equivalent to the following optimization problem

$$\min_{C} \frac{1}{2}\|\tilde{\boldsymbol{Z}} - \tilde{\boldsymbol{C}}\|_F^2 + P_1(\tilde{\boldsymbol{C}}) + P_2(\tilde{\boldsymbol{C}}), \tag{2.33}$$

where

$$\begin{aligned} P_1(\tilde{\boldsymbol{C}}) &= \|\tilde{\boldsymbol{\Lambda}}\tilde{\boldsymbol{C}}\|_{2,1}, \\ P_2(\tilde{\boldsymbol{C}}) &= \iota(\tilde{\boldsymbol{C}} = \boldsymbol{U}\Phi, \ \forall\Phi) = \begin{cases} 0 & \tilde{\boldsymbol{C}} = \boldsymbol{U}\Phi, \\ \infty & o.w.. \end{cases} \end{aligned}$$

We treat $\tilde{\boldsymbol{C}}$ in $P_1(\tilde{\boldsymbol{C}})$ and $P_2(\tilde{\boldsymbol{C}})$ as two different variables $\tilde{\boldsymbol{C}}_1 = \tilde{\boldsymbol{C}}, \tilde{\boldsymbol{C}}_2 = \tilde{\boldsymbol{C}}$. The dual problem

for equation (2.33) with dual variables $\Gamma \in \mathbb{R}^{\tilde{I}\times p}$, $\Upsilon \in \mathbb{R}^{\tilde{I}\times p}$ introduced for $\tilde{C}_1$, $\tilde{C}_2$ is

$$\min_{\Gamma,\Upsilon} \frac{1}{2}\|\tilde{Z} - \Gamma - \Upsilon\|_F^2 + P_1^*(\Gamma) + P_2^*(\Upsilon). \tag{2.34}$$

Since equation (2.33) is strongly convex, strong duality holds true. We can solve the primal problem through updating primal and dual iteratively. The final algorithm is summarized as below

Algorithm 5 Dykstra's prpjection algorithm

Initialize $\Gamma^{[0]} = \Upsilon^{[0]} = 0$, and $\tilde{C}^{[0]} = \tilde{Z}$

for $k = 0, 1, 2, \ldots$ **do**

$\tilde{C}_1^{[k+1]} = prox_{P_1}(\tilde{C}_k^{[k]} + \Gamma^{[k]})$,

where $prox_{P_1}(\boldsymbol{A}) = \{prox_{P_1}(\boldsymbol{a}_i)\}_{i=1}^{\tilde{I}} = \{(1 - \frac{\tilde{\lambda} w_I}{\|\boldsymbol{a}_i\|_2})_+ \boldsymbol{a}_i\}_{i=1}^{\tilde{I}}$ for any matrix $\boldsymbol{A} = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m)^T \in \mathbb{R}^{m\times r}$

$\Gamma^{[k+1]} = \Gamma^{[k]} + \tilde{C}_2^{[k]} - \tilde{C}_1^{[k+1]}$

$\tilde{C}_2^{[k+1]} = prox_{P_2}(\tilde{C}_1^{[k+1]} + \Upsilon^{[k]})$

where $prox_{P_2}(\tilde{C}) = \boldsymbol{U}\boldsymbol{U}^T\tilde{C}$

$\Upsilon^{[k+1]} = \Upsilon^{[k]} + \tilde{C}_1^{[k+1]} - \tilde{C}_2^{[k+1]}$

Stop until $\|\tilde{C}_2^{[k+1]} - \tilde{C}_1^{[k+1]}\|_\infty$ is small enough

2.4 Theoretical Properties

2.4.1 Minimax Lower Bound

In this section, we will show that the minimax lower bound for HSA method is of the order $[Jp + n\log J]/K$ where K can be set as $\|X\|_{2,\infty}$. We summarize it in the following results.

Theorem 2.4.1. *Let*

$$\boldsymbol{B}^* \in \mathcal{S}(J) = \{\boldsymbol{B} \in \mathbb{R}^{n\times p} : \|\boldsymbol{B}\|_{2,C} \le J\}$$

where $n \geq J \geq 2$, Then there exists positive constant C such that

$$\inf_{\hat{\boldsymbol{B}}} \sup_{\boldsymbol{B}^* \in \mathcal{S}(J)} \frac{\mathbb{E}\|\hat{\boldsymbol{B}} - \boldsymbol{B}^*\|_F^2}{[Jp + n \log J]/\|\boldsymbol{X}\|_{2,\infty}^2} \geq C > 0, \tag{2.35}$$

where $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^T$ is the predictor matrix, $\hat{\boldsymbol{B}}$ is any estimator and C is a universal constant.

Therefore, when $\boldsymbol{x}_i \overset{i.i.d.}{\sim} N(0, \tau^2\Sigma)$, $\|\boldsymbol{X}\|_{2,\infty}^2$ is of the order $\tau^2 p$ on average. If we assume a numerical setup with $\tau^2 : O(1)$, then $\|\boldsymbol{X}\|_{2,\infty}^2$ is of the order $O(p)$. In a balanced grouping case, it gives $\frac{n}{J}$ rows for each group. Denote $\boldsymbol{\beta}^* = ((\boldsymbol{\beta}_{I_1}^*)^T, \ldots, (\boldsymbol{\beta}_{I_J}^*)^T)$, $\hat{\boldsymbol{\beta}} = (\hat{\boldsymbol{\beta}}_{I_1}^T, \ldots, \hat{\boldsymbol{\beta}}_{I_J}^T)$. Then

$$\|\hat{\boldsymbol{B}} - \boldsymbol{B}^*\|_F^2 = \frac{n}{J}\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2^2.$$

As a result, the actual rate should be

$$[Jp + n \log J]\frac{J}{np} = O(\frac{J^2}{n} + \frac{J \log J}{p}).$$

It indicates when $n >> J^2$, $p >> J \log J$, we will have a small error. So large p is a blessing to HSA method in additional to the large sample size. Appendix A.4 gives the detailed proof of Theorem 2.4.1.

2.4.2 Prediction Upper Bound

In this section, we will show that the prediction error bound for an l_0-type penalty problem is of the order $p + (n+1)(n - \frac{n}{J^*})$ where J^* is the group number of the true model. We will use the logistic regression with Bernoulli distribution as out setup.

Given $\boldsymbol{X} \in \mathcal{R}^{n\times p}$, $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_n)^T$ with $\eta_i = \boldsymbol{x}_i^T \boldsymbol{\beta}_i$, let

$$\bar{\boldsymbol{X}} = \begin{pmatrix} \boldsymbol{x}_1^T & 0 & \ldots & 0 \\ 0 & \boldsymbol{x}_2^T & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \boldsymbol{x}_n^T \end{pmatrix} = diag(\boldsymbol{x}_i^T)$$

and the vectorization of $\boldsymbol{B}$ as $\bar{\boldsymbol{\beta}} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_n)^T \in \mathcal{R}^{np\times 1}$. Assume that

$$l_0(\boldsymbol{\eta}) = l_0(\bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}} \mid \boldsymbol{y}) = -\langle \boldsymbol{y}, \boldsymbol{\eta}\rangle_2 + \langle 1, \tau(\boldsymbol{\eta})\rangle_2 = \log P_{\bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}}(\boldsymbol{y}), \tag{2.36}$$

where $\tau(\boldsymbol{\eta}) = (\tau(\eta_1), \ldots, \tau(\eta_n))^T$ and $\tau(\eta_i) = \log(1 + \exp \eta_i)$ for any $i = 1, \ldots, n$. The loss corresponds to the Bernoulli distribution with cumulant function $\tau(\cdot)$.

Our tool to tackle the loss function is the *generalized Bregman* defined for any given differentiable ψ

$$\Delta_\psi(\boldsymbol{\alpha}, \boldsymbol{\beta}) := \psi(\boldsymbol{\alpha}) - \psi(\boldsymbol{\beta}) - \langle \nabla\psi(\boldsymbol{\beta}), \boldsymbol{\alpha} - \boldsymbol{\beta}\rangle_2. \tag{2.37}$$

If ψ is also strictly convex, $\Delta_\psi(\alpha, \beta)$ becomes the standard Bregman divergence $D_\psi(\boldsymbol{\alpha}, \boldsymbol{\beta})$, but our analysis does not require this for $l_0(\cdot)$. When $\psi(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_2^2/2$, $\Delta_\psi(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \|\boldsymbol{\alpha} - \boldsymbol{\beta}\|_2^2/2$, abbreviated as $\mathrm{D}_2(\boldsymbol{\alpha}, \boldsymbol{\beta})$.

Consider the l_0 penalty: $P(\bar{\boldsymbol{\beta}}) = \sum_{1\le i<j\le n} 1_{\|\boldsymbol{\beta}_i - \boldsymbol{\beta}_j\| \neq 0}$. We have the following theorem

Theorem 2.4.2. *Let* $\lambda_0 = \frac{p+n}{n}$. *Assume the following regularity condition holds:* $\exists \mu > 0$, $k \ge 0$ *such that*

$$\Delta_{l_0}(\bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}_1, \bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}_2) + K\lambda_0^2 P(\bar{\boldsymbol{\beta}}_1) + K\lambda_0^2 P(\bar{\boldsymbol{\beta}}_2) \ge \mu \mathrm{D}_2(\bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}_2, \bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}_1) \tag{2.38}$$

for all $\bar{\boldsymbol{\beta}}_1, \bar{\boldsymbol{\beta}}_2 \in \mathcal{R}^{np\times 1}$. *Let* $\lambda = A(\sqrt{K} \vee \frac{1}{\sqrt{\mu}})\lambda_0$ *with A to be a sufficiently large constant, Let*

$\hat{\boldsymbol{\beta}}$ be the optimal solution to

$$\min l_0(\bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}) + \frac{\lambda^2}{2} \sum_{1\le i<j\le n} 1_{\|\boldsymbol{\beta}_i - \boldsymbol{\beta}_j\| \neq 0}. \tag{2.39}$$

Then

$$\begin{aligned} \mathbb{E}[\|\bar{\boldsymbol{X}}\hat{\bar{\boldsymbol{\beta}}} - \bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}^*\|_2^2] &\le C \cdot \frac{K\mu \vee 1}{\mu^2}[p + \frac{p+n}{n} P(\hat{\boldsymbol{\beta}}^*)] \\ &= C \cdot \frac{K\mu \vee 1}{\mu^2}[p + (p+n)(n - \frac{n}{J^*})] \end{aligned} \tag{2.40}$$

where $\bar{\boldsymbol{\beta}}^*, J^*$ are the coefficient vector and group number of the true model, C is a universal constant.

The proof of the theorem can be found in Appendix A.5. Note that the regularity condition holds for linear regression with $K = 0$ and $\mu = 1$. For other penalties, we can obtain a similar error bound. But we will not illustrate in detail here. Specifically, according to the theorem, when $J^* = 1$, the upper bound error rate is of the level p, which equals to the number of free parameters in the model. The rate of the upper bound matches the minimax lower bound up to a multiplicity constant in the imbalanced grouping case, and the error rate is larger in the balanced grouping case. This results indicates that the uniform λ might generate relatively large error in some cases. Thus, it motivates us to use the data dependent weight w_{ij} to further reduce the error. Numerical results we will show later also indiates the effectiveness of properly constructed weights on the grouping structure.

2.5 Simulation Experiments

In this section, we will conduct series of experiments to test the performance of HSA method.

2.5.1 Running Time Comparisons

We first compare the running time of the three algorithms proposed in Section 2.3.2. Assume the total sample size is n, and half of the samples come from one of two logistic regressions described below:

$$logit(p_i) = \log \frac{p_i}{1-p_i} = \begin{cases} -0.5 - 0.1x_{1i} + 0.4x_{2i} & I_i = 1, \\ 0.2 + 0.5x_{1i} - 0.1x_{2i} & I_i = 2, \end{cases} \tag{2.41}$$

where I_i is the true state of sample i. The average sample variance ($\sum_{i=1}^{n} \frac{p_i(1-p_i)}{n}$) of the data is around 0.1. The 3D-scatter plot of $\boldsymbol{y}$ v.s. $\boldsymbol{X}$ is shown in Figure 2.2.

We choose different sample sizes $n = 50, 100, 150$ to make comparisons. We use the principle component induced ranking introduced in Section 2.2.4 to formulate the weight $\boldsymbol{w}$, and compared the running time of three algorithms when the solution of equation (3.5) converge to 2 groups. In order to make a fair comparison. We use the loss function value of Dykstra's projection algorithm as a reference, and stop the other two when their loss function values become lower than it. We repeat the experiments for 50 times. Figure 2.3 shows the comparison results of 50 repeated experiments. The x-axis stands for the sample size, and the y-axis stands for the quantile value of the running time in unit of second. Three plots correspond to ADMM, AMA and Dykstra's projection algorithm from left to right correspondingly. It can seen that ADMM runs fastest among the three, and Dykstra's projection algorithm is the slowest. The reason is because Dykstra's projection algorithm needs to have the same initialization value at each iteration in the inner loop, while the other two could have arbitrary initialized values. In this way, we can use the last updated value as the current initialization in the inner loop as a warm start. It makes the other two algorithms much faster than Dykstra's projection algorithm. As a result, we will use AMA for the rest experiments in the thesis.

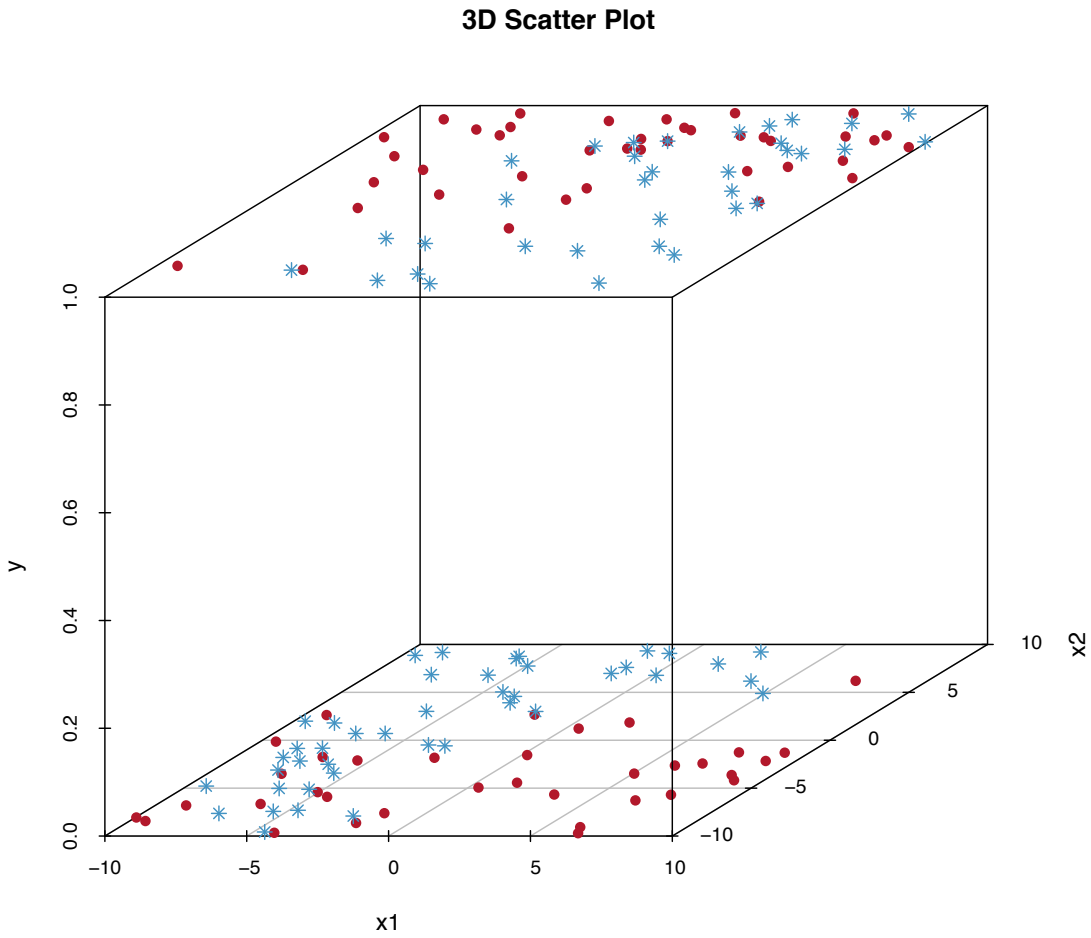


Figure 2.2: Running Time Comparison - 3D Scatter Plot of $\boldsymbol{y}$ v.s. $\boldsymbol{X}$.
The response variable is binary with each category containing samples from both groups. In another word, it has overlaps within each binary response value. The average sample variance is around 0.1. It is not trivial to separate them.

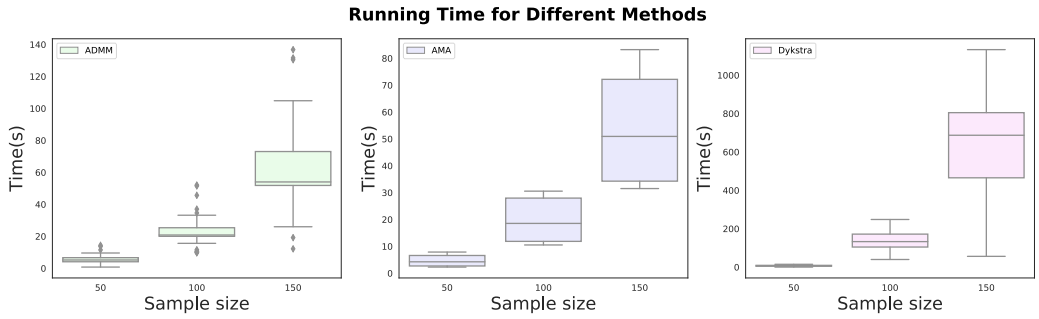


Figure 2.3: Running Time Comparison - Time Comparison Results.

2.5.2 Weighting Scheme Comparison under Different Noise Levels

Since we proposed two types of weighting schemes in Section 2.2.4, we will compare the performance of them through numerical examples. We first consider a two-state linear regression with sample size $n = 150$.

$$y_i = \begin{cases} -0.5 - 0.1x_{1i} + 0.4x_{2i} + \epsilon_i & I_i = 1, \\ 0.2 + 0.5x_{1i} - 0.1x_{2i} + \epsilon_i & I_i = 2. \end{cases} \tag{2.42}$$

where $\epsilon_i \sim N(0, \sigma^2)$ $(i = 1, \ldots, n)$. We chose multiple σ values to see whether the noise level will affect the performance or not. The total sample size is 150 with two equal size groups. The 3D-scatter plot of y v.s. X at $\sigma = 0.3$ is shown in Figure 2.4.

We conduct HSA using nearest neighborhood method and Bayesian method correspondingly to equation (3.4). We use rand index (Rand 1971a) to measure the similarity between the grouping results and the true label. Rand index ranges from 0 to 1, with 0 indicating that the two data clusterings do not agree on any pair of points and 1 indicating that the data clusterings are exactly the same. Figure 2.5 shows the rand index of the grouping results under different noise levels under 50 repeated experiments. It can be seen that Bayesian method outperforms nearest neighborhood method in this case, and the performance will not vary a lot as noise level increases from 0.01 to 0.3. That is a good sign indicating our method is relatively robust to the noise levels within the data.

2.5.3 Linear Regression and Logistic Regression Examples

Linear Regression: We first consider the linear regression example described in equation (3.4) with $\sigma = 0.3$. We use Bayesian method and get the principle component ranking described in Section 2.2.4. We set $k = 2.5$ in equation (2.15), and get the grouping results at different sample sizes ($n = 50, 100, 150, 200$). We also calculate the rand index of the "ideal" condition. Namely, calculate the distance of each sample to the hyper-plane

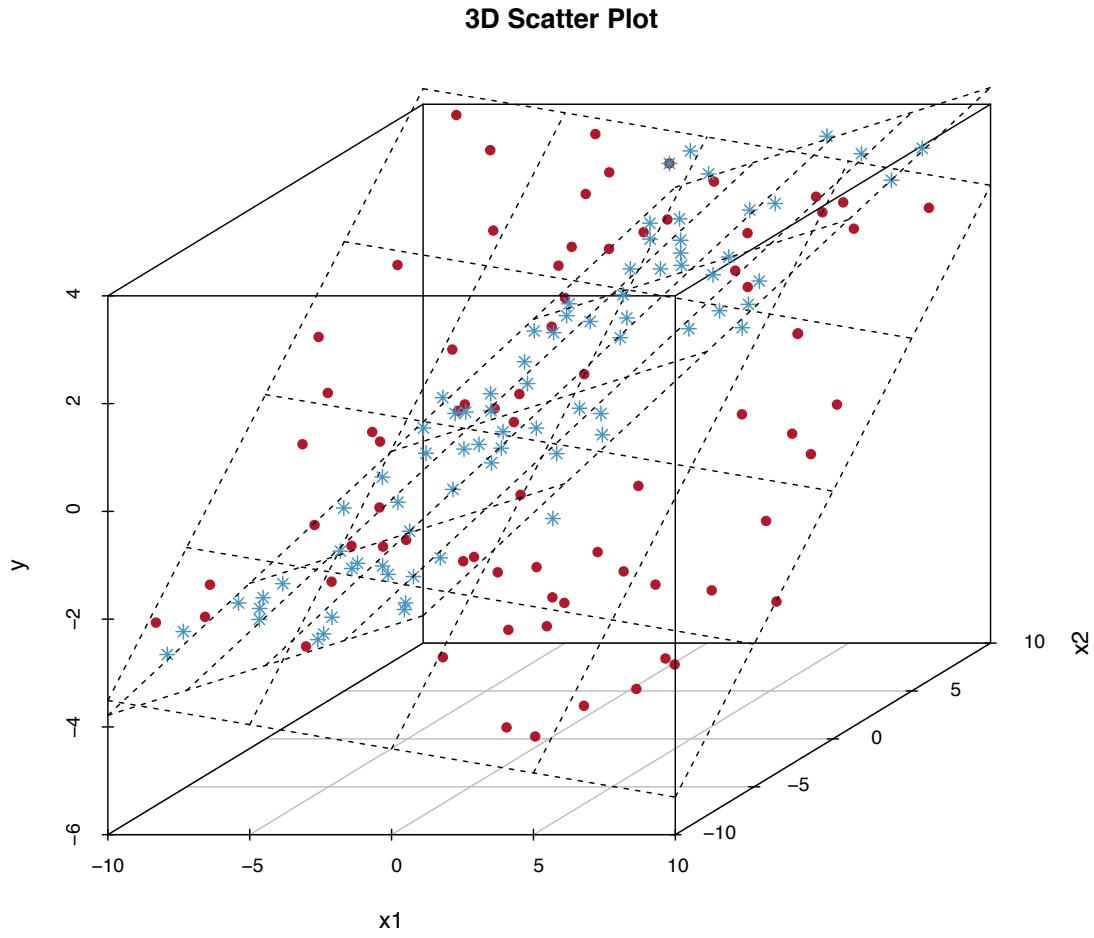


Figure 2.4: Weighting Scheme Comparison - 3D Scatter Plot of y v.s. X.
Two groups are marked with green triangular dots and blue dots. It can be seen that the two groups have a lot of overlaps, and it is not trivial to separate them.

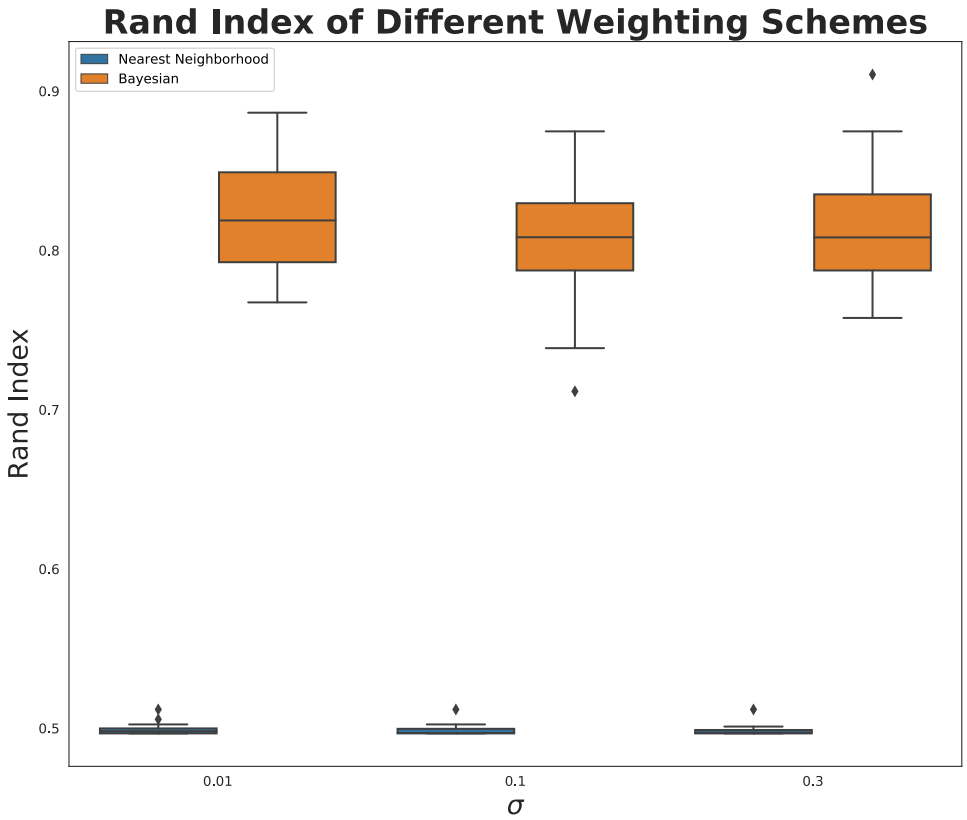


Figure 2.5: Weighting Scheme Comparison - Rand Indices for Different Weighting Schemes.

Bayesian method has much better performance in this setting. When the noise level increases, the performance does not get affect seriously. It indicates that HSA method is relatively robust to noisy data.

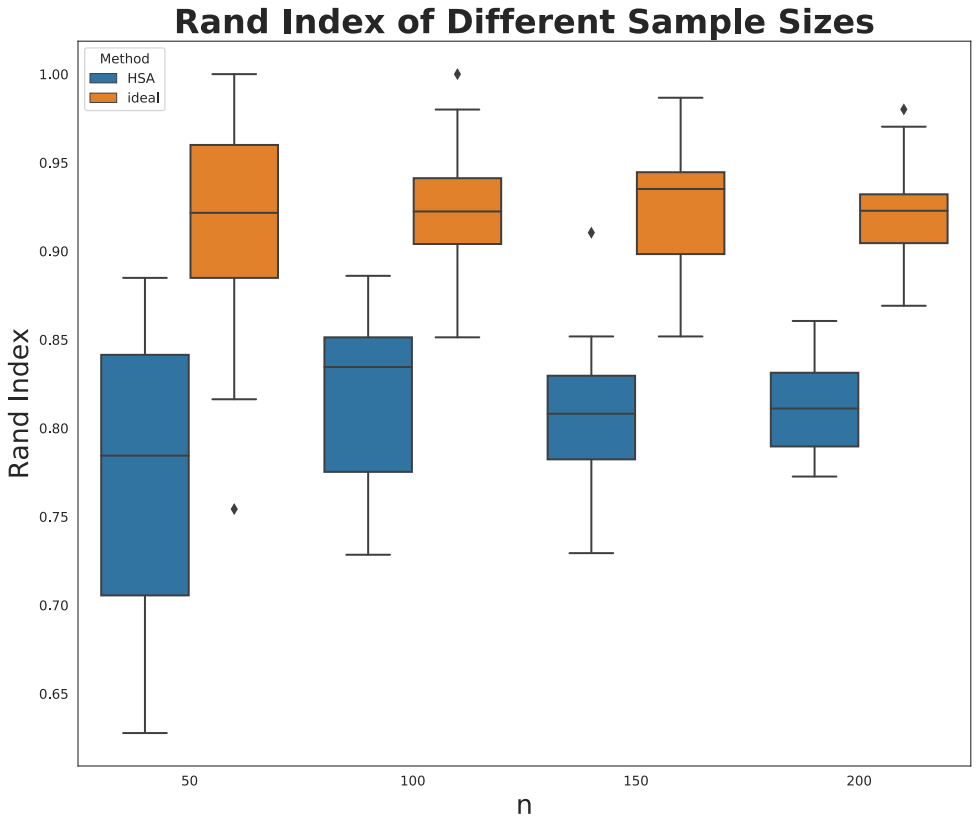


Figure 2.6: Linear Regression - Rand Indices under Different Sample Sizes.
It can be seen that the quantile value of the HSA method almost stay beneath the 25% quantile value of the ideal cases. As the sample size increases, the variance of the rand indices gets smaller.

determined by the true model of the two groups:

$$d_j = \frac{|\boldsymbol{x}_i^T \boldsymbol{\beta}_j - y|}{\|\boldsymbol{\beta}_j\|_2}, \quad j = 1, 2.$$

Then use the group with smaller distance as the "ideal" label for this sample. Figure 2.6 shows the rand index of HSA results and "ideal" label under 50 repeated experiments. It can be seen that the median values of the HSA rand index stay around 0.8, which is smaller than the "idea" results. As sample size increases, variance of rand indices gets smaller. It indicates that large sample size is potentially beneficial to HSA method.

Logistic Regression: We next consider the logistic regression example described in equation (2.41). Similarly, we use Bayesian method and get the principle component ranking described in Section 2.2.4. We set $k = 3$ in equation 2.15, and get grouping results at different sample sizes ($n = 50, 100, 150, 200$). We also calculate the group label generated from the "ideal" condition. Namely, calculate the log-likelihood value of each observation using the true coefficients of group 1 and group 2:

$$loglike_j(\boldsymbol{x}_i, y_i, \boldsymbol{\beta}_j^*) = -y_i \cdot \boldsymbol{x}_i^T \boldsymbol{\beta}_j + \tau(\boldsymbol{x}_i^T \boldsymbol{\beta}_j), \quad j = 1, 2.$$

Then use the smaller one as its group. Figure 2.7 shows the rand index of HSA and "idea" label under 50 repeated experiments. It can be seen that the overall performance of the logistic regression is worse than linear regression because of the stochastic random noise within the Bernoulli distribution. HSA's rand index is a little lower than the "ideal" case. On the other hand, as sample size increases, the variance of the rand index gets smaller. It also indicates that large sample size might be beneficial to the clustering performance.

In summary, we successfully capture the hidden structure within a data in both linear regression and logistic regression examples using HSA. The performance of linear regression is better than logistic regression as expected. In the next section, we will apply HSA method to real data.

2.6 Real Data Examples

This section illustrate two real data applications of HSA method.

2.6.1 Tourism Data

The first application focus on $n = 180$ monthly data concerning *tourists overnights* (X data in millions) and *attendance at museums and monuments* (y data in millions) in Italy over the 15-year period spanning from January 1996 to December 2010. The data have been

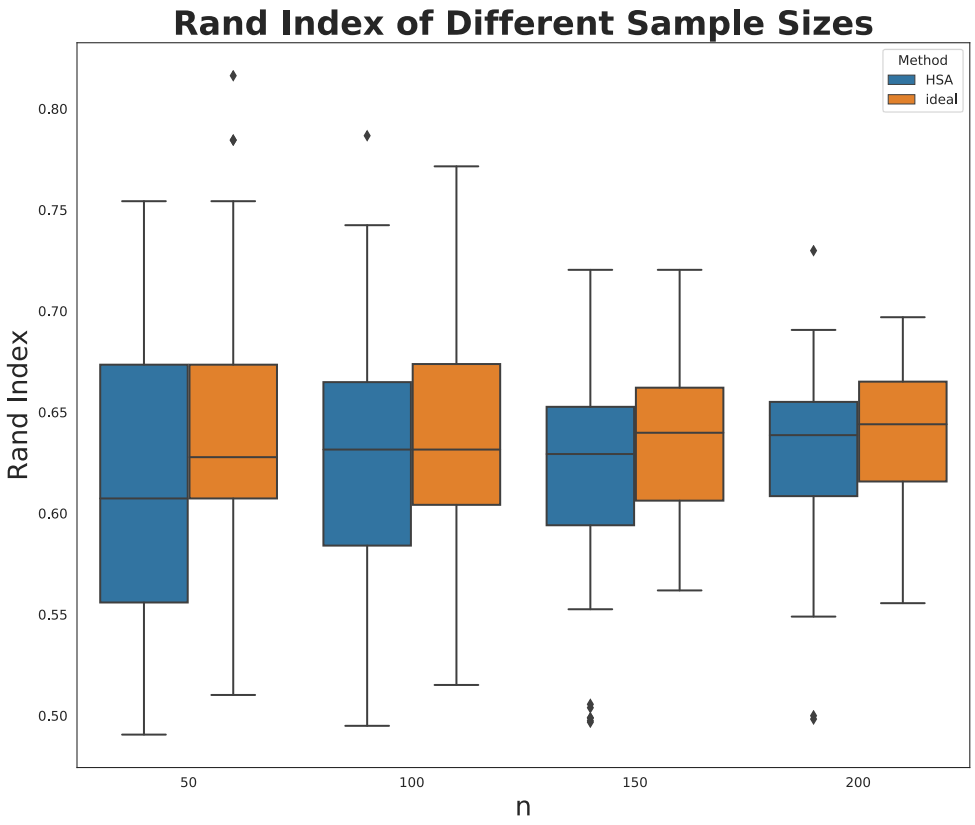


Figure 2.7: Logistic Regression - Rand Indices under Different Sample Sizes.
The overall rand indices of logistic regression is smaller than linear regression. But the rand indices of HSA is comparable with the "idea" case now. It indicates that clustering for logistic regression is more difficult than clustering linear regression due to the stochastic randomness within the Bernoulli distribution. As the sample size increases, the variance of the rand indices gets smaller.

analyzed by Cellini and Cuccia (2013) and Ingrassia, Minotti, and Punzo (2014) and are available at *http://docenti.unict.it/punzo/Data.htm*. Scatter plot of $\boldsymbol{y}$ v.s. $\boldsymbol{X}$ in Figure 2.9 shows some heterogeneity condition which indicates some hidden group-structure. Thus, we will use HSA with respect to linear regression to explore this data.

Since it is a simple linear regression, instead of using Bayesian method to get a posterior estimation of the coefficient matrix, we can directly use $[\boldsymbol{X}, \boldsymbol{y}]$ to generate the ranking distance using equation (3.9). We set $k = 3$, and perform HSA on scaled $\boldsymbol{y}$ and $\boldsymbol{X}$ without the intercept. Figure 2.8 shows the histogram of grouping results up to 5 different groups. The x-axis stands for different months. It can be seen that the grouping structure align with months information quite well. The five groups mainly correspond to the following month:

Group 1: January, February, November and December;

Group 2: March and October;

Group 3: April and May;

Group 4: June and September;

Group 5: July and August.

The scatter plot of grouping results in Figure 2.9 give us a straightforward visualization of the pattern changing process while the number of groups changes. From these two figures, it can be seen that we successfully find out the hidden structure within the data corresponding to the month information.

2.6.2 Bank Marketing Data

The second data is related with direct marketing campaigns of a Portuguese banking institution. The goal is to predict if the client will subscribe a term deposit (y). The data has been analyzed by Moro, Cortez, and Rita (2014), and can be accessed through UCI machine learning repository (*http://archive.ics.uci.edu/ml/datasets /Bank+Marketing*). We use the smallest data set: bank.csv with sample size $n = 4521$. The data included 20 attributes, but the analysis we present below only concerns the following

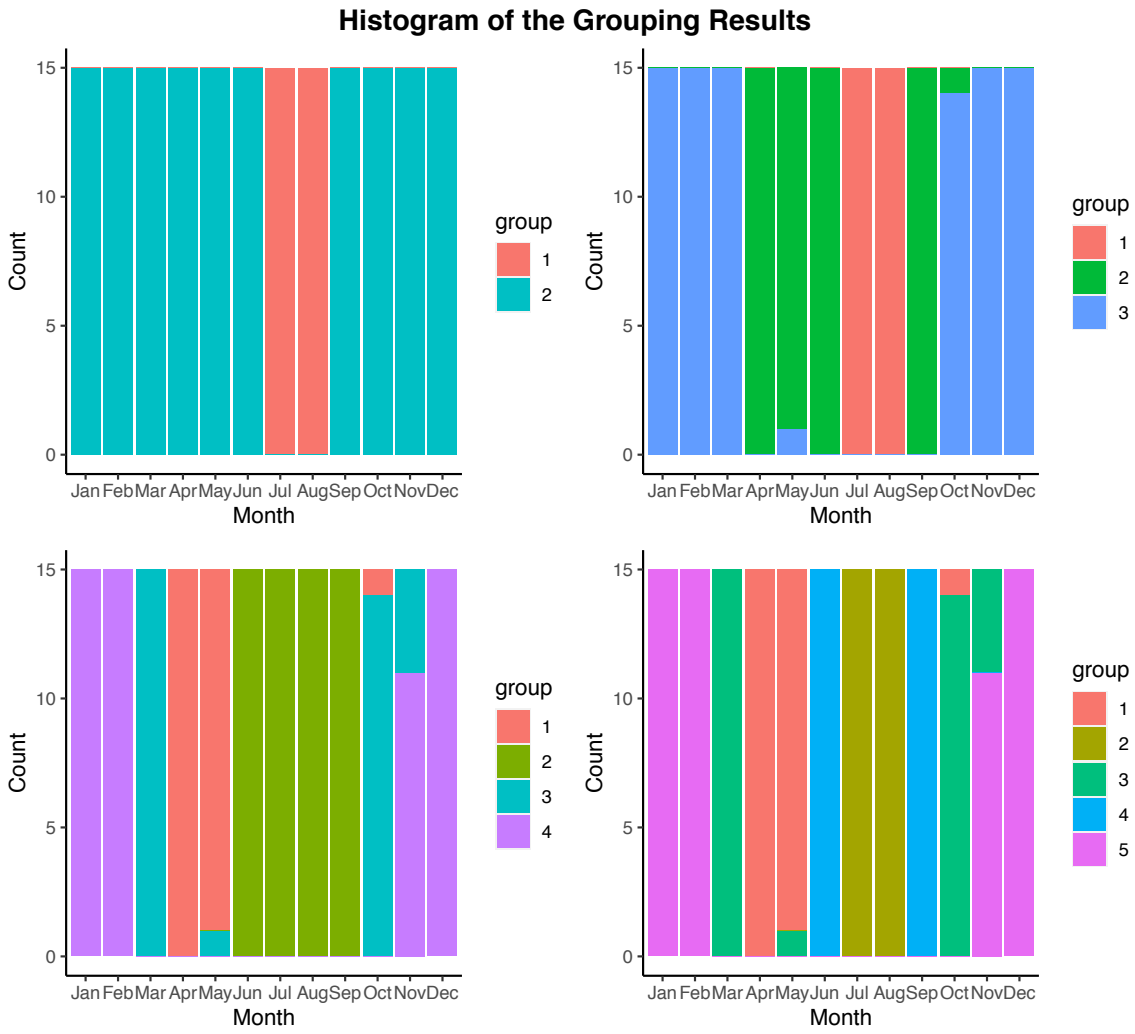


Figure 2.8: Tourism Data - Monthly Histogram.
When $J = 2, 3, 4, 5$, the grouping results align with the unknown variable: month information quite well.

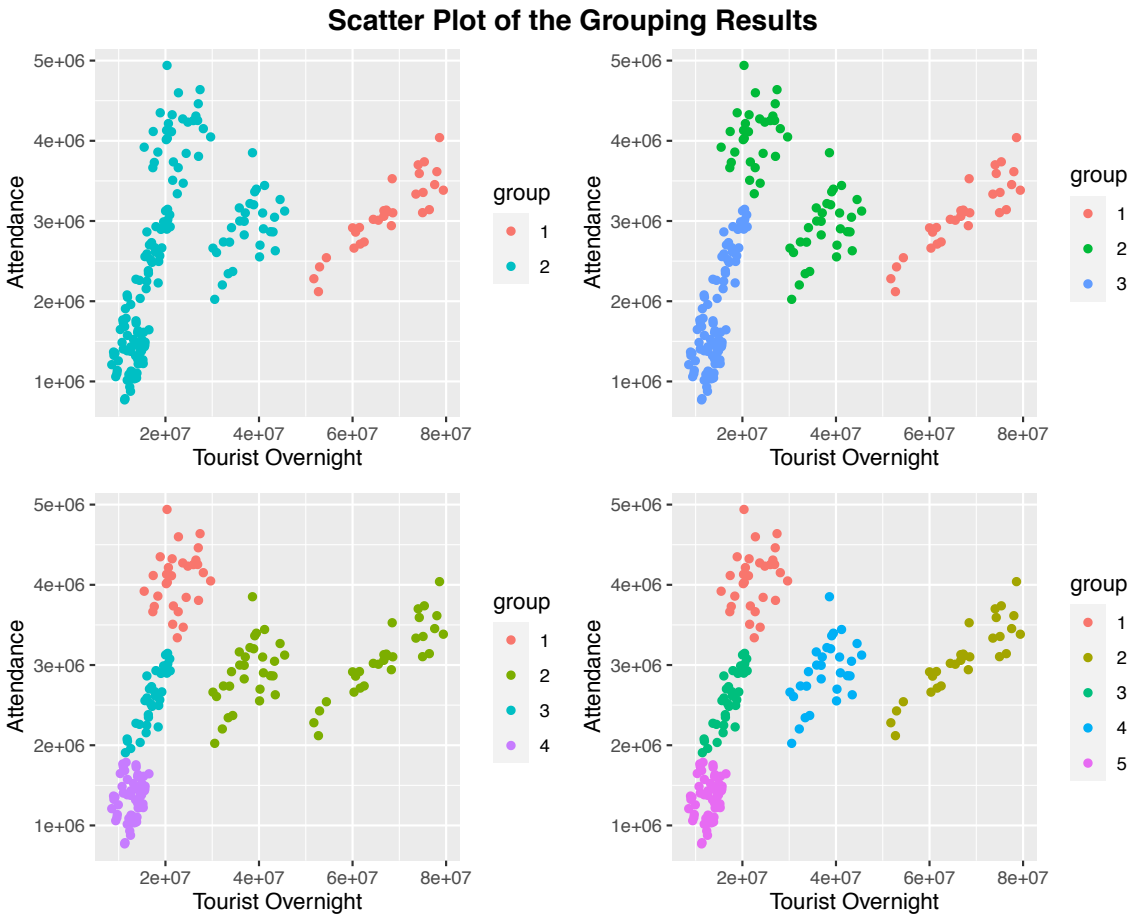


Figure 2.9: Tourism Data - Clustered Scatter Plot.
This figure contains the grouping results with $J = 2, 3, 4, 5$. It can be seen that as group number increases, samples which deviates from the other will gradually split out, and formulate a new group. Some grouping mis-match might happen during this process because $\boldsymbol{B}$ keeps changing slightly. In the end, the five groups confirm with our intuition.

Table 2.1: Coefficients of the Original Logistic Regression

	Estimates	Std. Error	P value
(Intercept)	0.057348	0.027161	0.034791
age	0.001131	0.000465	0.015062
education	0.022385	0.006181	0.000296
housing	-0.061664	0.009726.	2.52e-10

subset of variables:

Age: age of the client;

Education: 0: “unknown”, 1: “primary”, 2: “secondary”, 3: “tertiary”;

Housing: 1: “yes”, 0: “other (no or unknown)”.

We consider a Logistic regression with response variable $\boldsymbol{y}$ versus three predictors above. Coefficients estimation of the model is summarized in Table 2.1.

We can use HSA method to see whether there exists hidden structure within the data. Sometimes, due to the limited computational power, it is not feasible to conduct HSA HSAon the whole data set together without GPU enabled computing environment. As a result, we introduce a model-based stratified sampling method to divide the whole data set into different sub-blocks, then conduct HSA process on each sub-block to get the final group pattern. We will justify the effectiveness and robustness of the result at the same time.

The whole idea of model-based stratified sampling can be summarized into a diagram shown in Figure 2.10.

1. Divid the whole data set into K different parts with similar data structure. Since the response variable is binary, we perform k-means clustering to each binary value, and split the clustered data into K equal parts correspondingly. The process is shown in the left penal in the diagram.

2. Combine the equal parts from different clusters into one sub-block. As a result, we can get K different sub-blocks in total as shown in the middle part in Figure 2.10.

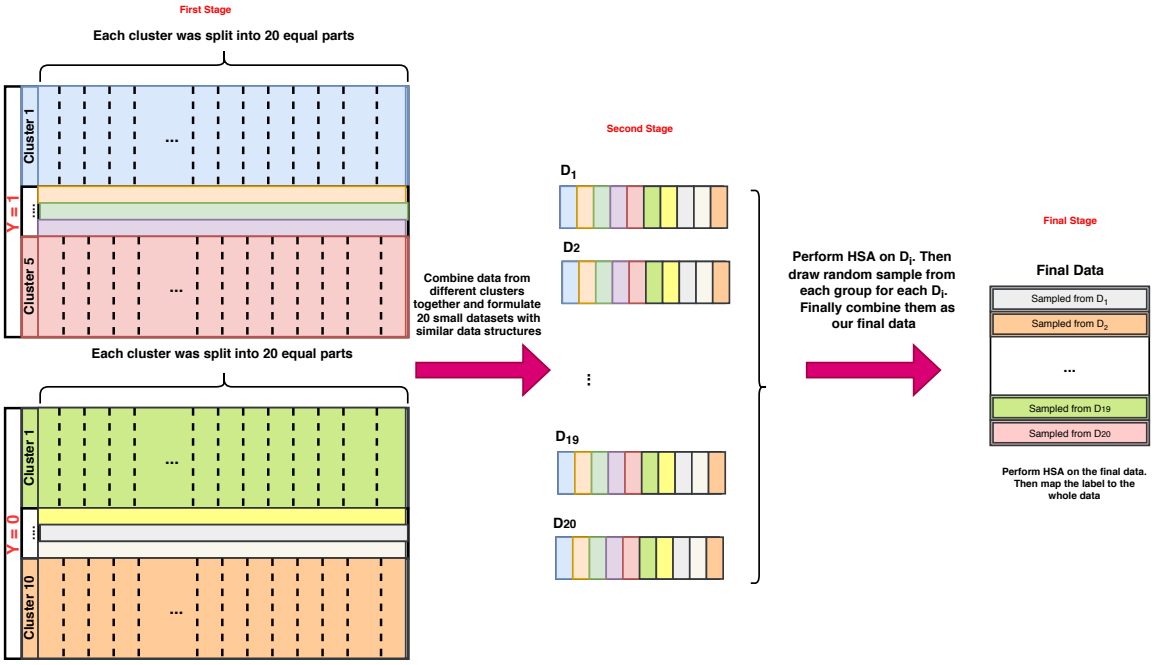


Figure 2.10: Model-based Stratified Sampling - Diagram.

We perform HSA process on each sub-block, and get the state estimates from each block. In this way, samples from the same state would be more similar to each other in one sub-block.

3. Once getting the states estimates for each sub-block, we can draw samples from each state of each block. In order to construct a mapping which we would use later, the sub-sampling process should be performed based on a clustering procedure for each state in a block. For example, we can divide the samples from one state of a sub-block into k clusters, and draw 1 sample from each clusters. Then we get k subsample from this state. In this way, we can map the state of these k subsamples to other samples in this state naturally. Finally, we combine all subsamples from different sub-blocks into a smaller data set. This data set could be very representative to the overall data structure. As a result, we can make HSA on the it and map the resulted label to the whole data according to their sub-sampling source. In this way, all samples would get a label from conducting HSA on the final small data.

Table 2.2: Coefficients of Two Separated Logistic Regression.

		Estimates	Std. Error	P value
1st Group	(Intercept)	−0.0498972	0.0168647	0.00313
	age	0.0003781	0.0002909	0.19373
	education	0.0197711	0.0039407	$5.73e-07$
	housing	0.8742960	0.0098088	$< 2e-16$
2nd Group	(Intercept)	0.9444135	0.0119306	$< 2e-16$
	age	0.0001651	0.0001879	0.379600
	education	0.0084764	0.0024500	0.000549
	housing	−0.9626009	0.0056923	$< 2e-16$

Using the procedure above, we chose $K = 20$ in our data, and got 20 different sub-blocks in step 2. The total number of groups is set as 2 and 3 respectively. In order to testify the robustness of the grouping results in terms of the sub-sampling procedure. We chose 10 different random seeds and calculated the rand index between different grouping results. Figure 2.11 shows the rand index with different number of groups. It can be seen that almost all values are higher then 0.9. This indicates that our sub-sampling method generate consistent results from different random seeds. Thus, the robustness of the proposed method is good.

Once we get the estimated states with two groups and three groups, we fit two and three separate Logistic regressions based on their labels. Table 2.2 shows the modeling results for two groups. It can be seen that in these two models, both of the age features are not significant, and the coefficients for the housing features have opposite signs. It might stands for two different types of people who manage their money with totally different habits.

As for the modes with three groups, we also fit three separate models based on the estimated states. Table 2.3 shows results of three groups. It can be seen that the age features are significant for two models, and one of the three groups has opposite sign for the housing and education feature compared with the other two groups.

In summary, from these two real data examples, we can see that HSA method can help

Table 2.3: Coefficients of Three Separated Logistic Regression

		Estimates	Std. Error	P value
1st Group	(Intercept)	$1.000e+00$	$5.108e-17$	$< 2e-16$
	age	$-5.238e-20$	$7.520e-19$	0.944
	education	$-8.577e-18$	$1.257e-17$	0.495
	housing	$-1.000e+00$	$2.515e-17$	$< 2e-16$
1st Group	(Intercept)	$1.000e+00$	$1.859e-16$	$< 2e-16$
	age	$-7.821e-18$	$3.165e-18$	0.0138
	education	$-9.952e-17$	$5.200e-17$	0.0562
	housing	$-1.000e+00$	$1.024e-16$	$< 2e-16$
3rd Group	(Intercept)	$5.469e-17$	$5.522e-18$	$< 2e-16$
	age	$-8.764e-19$	$9.518e-20$	$< 2e-16$
	education	$1.586e-18$	$1.295e-18$	0.221
	housing	$1.000e+00$	$3.273e-18$	$< 2e-16$

us identify some hidden structures within the data. But when the data involves complex information, or the data doesn't contain comprehensive information with respect to what we need, the hidden structure might not be easily seen as what we showed in the Tourism data. For example, in the Bank Marketing data, we cannot directly attribute the grouping pattern into some current available features. But fitting separate models according to the grouping labels obviously convey different information from each group, and that is an important sign of hidden structures. As a result, HSA method illuminates us to improve the original model by fitting separate models or adding additional features.

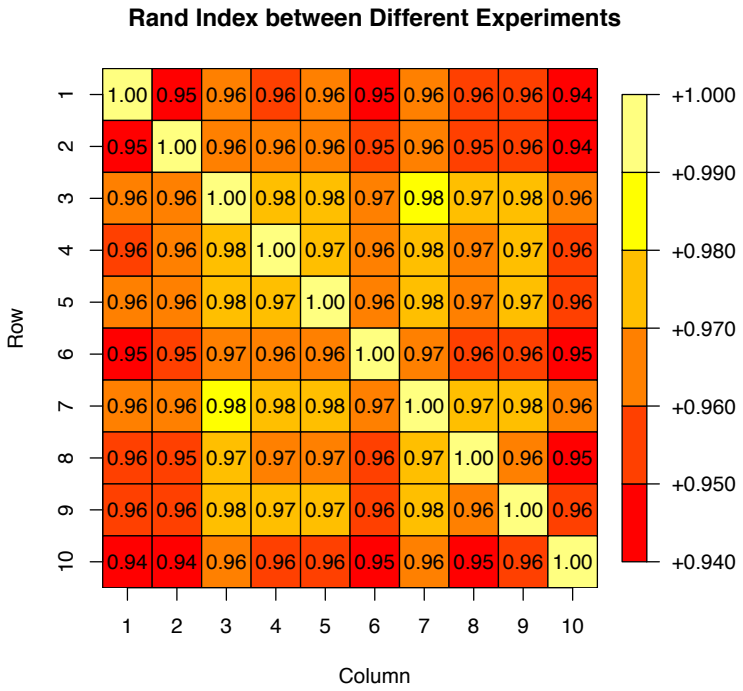
2.7 Conclusion

This chapter has proposed a relational based clustering model: HSA, to identify the latent variable or hidden structures within the data under generalized linear models. A convex optimization problem is formulated, and we proposed different algorithms to solve it. We select out the best algorithm based on the running time of simulation examples, and validate the performance of the model through numerical examples under different settings. We

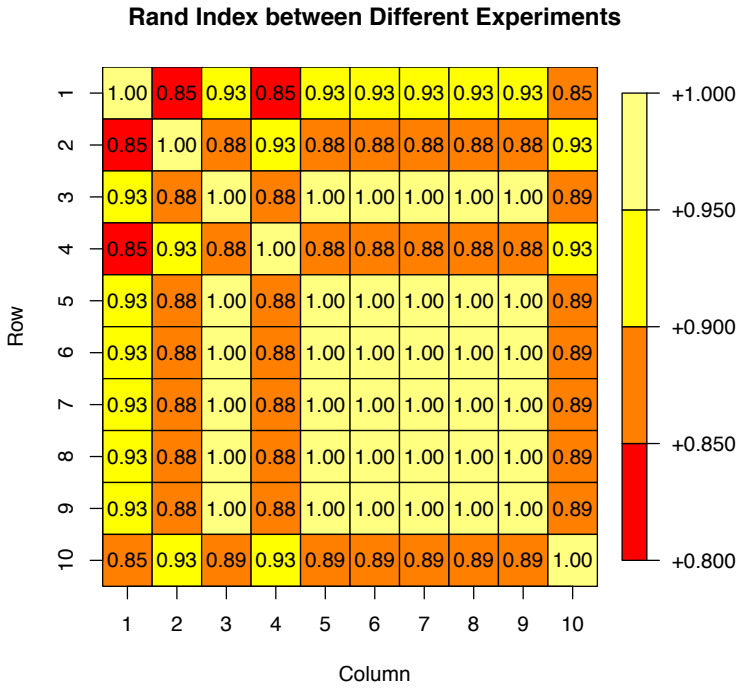
provide two real data examples. For the Italy tourism data, we successfully cluster out the subgroup aligning well with an additional feature: month, which is not included when we fit the model. For the second data, since the sample size is relatively large, we propose a model-based stratified sampling method to simultaneously conduct HSA on different sections of the data. The resulting clusters are robust to the stratified sampling. The data is clustered into two groups and three groups respectively, and the corresponding results are quite interesting and illuminating.

One potential drawback of HSA method is that the quality of the clustering suffers from its inability to perform adjustment, once the pairwise weight value is settled, the clustering pattern will not be changed afterward. It makes the weight choosing process extremely crucial to the overall performance of the model. This might lead to somewhat misleading clusters if the pairwise weight value is not properly chosen. In order to alleviate the strong dependency of the overall performance on the weighting schemes, we can try to use non-convex penalty terms (She 2012) to iteratively adjust the clustering patterns in the future.

In summary, through the simulation and real data examples, we can see the potentiality and effectiveness of HSA. But it is also worthy to note that the quality of the weighting scheme hugely affect the performance of model. Thus, in the next chapter, we will introduce a modeling framework based on HSA method. When there exists some structural variable that exclusively determines the state of each observation, HSA method will work better. The framework will utilize the power of structural variables to improve the model performance.



(a) Rand Index with Two Groups.



(b) Rand Index with Three Groups.

Figure 2.11: Rand Index between Different Random Seeds.

CHAPTER 3
HSA ENHANCED WITH STRUCTURAL VARIABLES

3.1 Introduction

When we analyze the relationship between a target variable and relevant features, we might find out their relationship cannot be easily described by a simple distribution potentially because of lots of reasons like missing latent variables or heterogeneity issues, etc.. As a result, we need clustering algorithms like HSA to split the data set into sub-populations to make the structure simple and clear. In many cases, such difficulty is because of the existence of structural variables that control the general structure of the model, instead of affecting the model as a standard covariate. Common examples of models based on this rationale would be a threshold or varying coefficient models in regression or functional coefficient models in time series. The variable that controls the change of coefficient is the structural variable. One typical example is the threshold AR (TAR) (Tong and Lim 2009) model, in which different AR models are assumed based on the regime that a threshold variable is in. Threshold variable driven switching AR models (TD-SAR) is a variation of TAR model. It was first proposed by Wu and Chen (2007), it combines the strong information provided by the observable threshold variable and potential randomness in the switching mechanism together. It takes advantages of both switching autoregressive (SAR) (Tong and Lim 2009) model and TAR model. In some more complicated cases, the hidden structure may not be easily captured by a simple threshold variable with some single regime structure. Instead, we need more advanced and more complex settings to extract decision boundaries or regions from the structural variable, and the structural variable could be in multi-dimensional space with non-linear decision boundaries. Thus, we need a powerful tool to identify the proper structural variable and recover the decision boundary. In

this chapter, we introduce a novel modeling framework: Structural variable Driven-HSA (SD-HSA), to accomplish this. SD-HSA framework assumes there exists some structural variable controlling the group structure. Once we get the estimated state labels for each data subject from HSA, we can build separate models based on the decision region of the structural variable.

Aside from the motivation of introducing the structural variable to characterize the relationship between the target response and predictors, we also found out that a proper structural variable could possibly enhance HSA method as well. Since HSA method specifies an objective function with a pairwise weight penalty, how to choose the weighting scheme is crucial to the grouping performance. We note that when the grouping is based on some structural variable, the information of the structural variable can be potentially very helpful by allowing the weight depend on the structural variable. This framework improves HSA method from an unsupervised data exploration tool to a well-defined system for building a better model with the help of structural variables. As a result, we can incorporate more potentiality into HSA.

We can use an example to help us understand the logic under SD-HSA. Assuming we want to predict stock prices from different companies using historical data. We could fit an autoregressive (AR) model on the stock prices. But we realize that some other features like each company's Fama-French factors might also affect the stock price. To be more specific, companies with different scales would be endowed with a different set of parameters for the AR models corresponding to their Fama-French factors. Thus, Fama-French is the structural variable in this case. We can use HSA method based on the structural variable to estimate the stock price of different companies. By constructing different models based on decision region generated by the structural variable, we can fit separate AR models and predict stock prices correspondingly.

In summary, we introduce a model framework to characterize the relationship between the response variable and relevant features under GLM's setting with some pre-chosen

structural variable. The framework mainly contains three stages. At the first stage, we narrow down the potential candidates of structural variables from a large size to a relatively small size to reduce the computational cost. At the second state, we use HSA to get the state estimation of each sample. The weight formulation incorporates the information obtained from the structural variable candidates. At the last stage, we choose the best model along with the respective structural variable by model selection criteria like AIC or BIC values. At last, we can recover the decision boundary based on the estimated states using classification methods like support vector machine, logistic regression, etc..

The rest of the chapter is organized as follow. We first specify the K-state GLM model, then introduce the modeling framework, SD-HSA, to obtain the model. We also provide simulation examples to testify the performance of the modeling framework. Finally, we apply our method on real data examples and compared with the results obtained from the method introduced in Wu and Chen (2007).

3.2 Model Specification: J-state GLM

Give a data set of n samples with response $\boldsymbol{y} = (y_1, \ldots, y_n)^T \in \mathbb{R}^n$ and features of interest $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^T \in \mathbb{R}^{n \times p}$, the J-state GLM model has the following form:

$$y_i \mid \boldsymbol{x}_i \sim GLM(\boldsymbol{\beta}^{[I_i]}), \quad 1 \leq i \leq n \tag{3.1}$$

where $I_i \in \{1, \ldots, J\}$ is the state variable for the i-th observation, and $\boldsymbol{\beta}^{[I_i]}$ is the corresponding coefficient vector. Assume there are J sub-populations within the data. Each population follows an independent GLM with a different set of parameters. The true label of I_i is determined by a pre-chosen m-dimensional structural variable $\boldsymbol{z}_i = (z_{1i}, \ldots, z_{mi})^T$ ($i = 1, \ldots, n$), and the vector space of the structural variable is divided into J different regions:

$R_1, \ldots, R_J$. As a result, the true state of the i-th sample is determined by

$$I_i = j, \quad \text{if } z_i \in R_j. \tag{3.2}$$

We can estimate the state of each sample using HSA with the help of the structural variable z_i, denote as $\hat{I}_i$. Once the state estimate is obtained, we can recover the decision boundary of z by treating the state estimate $\hat{I}_i$ as labels, z_i as predictors to make classification. Different algorithms like logistic regression, support vector machine, etc. can be adopted based on preferences of practitioner. Then we can naturally get a decision region formed by the structural variable, and fit separate models based on it.

3.3 SD-HSA: a Three-stage Modeling Framework

In order to obtain the J-state GLM, we construct a modeling framework involving three stages.

Stage one: narrowing down structural variable candidate pool

It is always a difficult task to determine appropriate structural variables for practitioners in practice. To find the structural variable, a commonly used method is to traverse all combinations of the possible structural variables, fit all the corresponding models, and find the best one according to model selection criteria such as Bayesian Information Criterion (BIC) or out-sample prediction performance. But when researchers begin to consider linear combination of several variables as the structural variable, the traditional exhaustive method is not sophisticated enough and might result in too much additional computation cost. Thus, we propose two approaches to narrow down the number of potential structural variables to quickly filter out promising structural variable candidates. Through the narrowing down process, we can quickly search among a large size of candidate variables, and make further analysis only on a much smaller number of structural variable candidates.

Stage two: HSA process

Once the number of structural variable candidates is reduced, we can use HSA method on each of them to make state estimates. The loss function of HSA method is constructed by two parts: the first part is the component-wise log-likelihood function of the GLM, the second part is a regularization term to enforce equi-sparsity. A weighted group l_1 penalty is used here. Specifically, the data dependent weight is determined by the structural variables based on some pre-defined ranking similarity.

Stage three: model selection and recovering the decision boundary

Once we get the state labels for each sample based on different structural variables, we can select the best model based on model selection criterions like AIC or BIC values. Then based on the estimated states selected from the best model, we can recover the decision boundary for the structural variable using classification methods like logistic regression, SVM etc.

In the next three sections, we will illustrate in more detail about the three stages.

3.3.1 Stage One: Narrowing Down Candidates of Structural Variables

In order to narrow down the possible structural variable candidates, we need to filter out promising variables which are more likely to be aligned with the coefficient structure within the data. For HSA method, since each sample is assigned with an independent coefficient vector $\boldsymbol{\beta}_i$ $(i = 1, \ldots .n)$, we can get a pilot estimate for $\boldsymbol{\beta}_i$ for each sample, denote as $\boldsymbol{\beta}_{i0}$. If the pilot estimation is reasonable, it should be helpful for us to filter out the ideal structural variable candidates by modeling the pilot estimation and structural variables together.

Intuitively, when two samples actually belong to the same group, their pilot estimation $\boldsymbol{\beta}_{i0}$s should also be close to each other. As a result, we can use the similarity between $\boldsymbol{\beta}_{i0}$s to select out the structural variables which align better with their pilot estimations.

For notation simplicity, we stack $\boldsymbol{\beta}_{i0}$s together into a matrix $\boldsymbol{B}_0 = (\boldsymbol{\beta}_{01}, \boldsymbol{\beta}_{02}, \ldots, \boldsymbol{\beta}_{0n})^T \in \mathbb{R}^{n \times p}$. We can estimate $\boldsymbol{B}_0$ using two approaches. The first approach is easy to conduct, but the performance is not very stable, especially in some difficult scenarios in which different

groups have overlaps. Thus, we propose the second method based on Bayesian analysis. It generates more trustworthy estimates in those difficult scenarios. We summarize the two methods as below:

Pilot estimation methods

Pilot estimation (i): initial HSA method

As we will show later, for HSA method, we get the parameter estimation matrix $\boldsymbol{B}$ through minimizing

$$\boldsymbol{B}^* = \arg\min_{\boldsymbol{B}} l(\boldsymbol{B} \mid \boldsymbol{y}, \boldsymbol{X}) + \sum_{1 \le i < j \le n} P(\boldsymbol{\beta}_i - \boldsymbol{\beta}_j; \lambda_{ij}), \tag{3.3}$$

where $l(\boldsymbol{B} \mid \boldsymbol{y}, \boldsymbol{X})$ is the negative log-likelihood function of the corresponding GLMs, and $P(\boldsymbol{\beta}_i - \boldsymbol{\beta}_j; \lambda_{ij})$ is the regularization term to enforce equi-sparsity. $\lambda_{ij} = \lambda w_{ij}$ is a regularization parameter to control the coalescence of the parameter estimation for each observation. When λ is small, we can get a coefficient matrix $\boldsymbol{B}$ having lots of distinct rows. Thus, we choose a small uniform λ_{ij} (λ is small and $w_{ij} = 1$). Let the corresponding solution of equation (3.3) to be the pilot estimation $\boldsymbol{B}_0$. This estimation is simple to obtain, and can reflect the structure of the data to some extent. Since this method is in a sense equivalent to fitting an independent model on each observation with a minor regularization, it can be imagined that the uncertainty of the resulting coefficient vector for each sample is relatively high. On the other hand, theoretical results in Chapter 2 show that uniform λ will generate solution with relatively large error rate in some cases. Thus, it might not be accurate enough to be our pilot estimate.

Pilot estimation (ii): Bayesian method

Since the observed value is actually a reflection of the true mechanism within the data, we can use Bayesian method to get the pilot estimation matrix $\boldsymbol{B}_0$. The whole process contains two steps:

Step one: Fit a standard GLM on the data. We generally recommend standardizing each

predictor to ensure all variables are on the same scale. The estimated coefficients from the GLM is denoted as $\hat{\boldsymbol{\beta}}$ with standard deviation $\hat{\sigma}_{\boldsymbol{\beta}}$.

Step two: For each sample $(\boldsymbol{x}_i, y_i)$ $(i = 1, \ldots, n)$, we can get an independent posterior estimation of $\boldsymbol{\beta}_i$ from GLM. The exponential likelihood function of y_i given $\boldsymbol{x}_i$ is described in equation (2.8). Bayesian analysis requires specifying prior distribution $f(\boldsymbol{\beta}_i)$. Based on the GLM from step one, it is natural to assume the prior distribution of $\boldsymbol{\beta}_i$ follow $\boldsymbol{\beta}_i \sim Normal(\hat{\boldsymbol{\beta}}, c\hat{\sigma}_{\boldsymbol{\beta}})$ where c is positive constant, and we usually set it larger than 1 to allow more flexibility to its posterior estimation. The posterior distribution of $\boldsymbol{\beta}_i$ satisfies:

$$f(\boldsymbol{\beta}_i \mid y_i, \boldsymbol{x}_i) \propto f(\boldsymbol{\beta}_i) f(y_i \mid \boldsymbol{x}_i), \tag{3.4}$$

where $f(y_i \mid \boldsymbol{x}_i)$ is the GLM likelihood function. We can draw this posterior distribution through Markov Chain Monte Carlo (MCMC) using Rstan (`https://mc-stan.org/docs/2_21/stan-users-guide/index.html`), and estimate $\boldsymbol{\beta}_i$ as the corresponding posterior mean. Since the estimation of the intercept is meaningless (because they are all estimated based on the same value 1), we drop it and denote the rest as $\hat{\boldsymbol{\beta}}_{0i}$ $(i = 1, \ldots, n)$. In this way, posterior estimations from each sample form a matrix $\boldsymbol{B}_0 = (\hat{\boldsymbol{\beta}}_{01}, \ldots, \hat{\boldsymbol{\beta}}_{0n})^T$. In this way, we obtain the pilot estimation matrix $\boldsymbol{B}_0$ properly accounting for the information contained in the data.

Once we get the pilot estimation matrix $\boldsymbol{B}_0$, we can conduct the narrowing down process in two ways:

Narrowing down methods

Narrowing down (i). Clustering based narrowing down process

Figure 3.1 shows the diagram of the clustering based narrowing down process. Based on the pilot estimations $\boldsymbol{\beta}_{i0}$s, we can use k-means clustering to get an pilot state estimate for each sample, denote it as $\hat{I}_{i0}$ $(i = 1, \ldots, n)$. Then we apply classification methods like support vector machine (SVM) or other appropriate methods with response as $\hat{I}_{i0}$ and structural

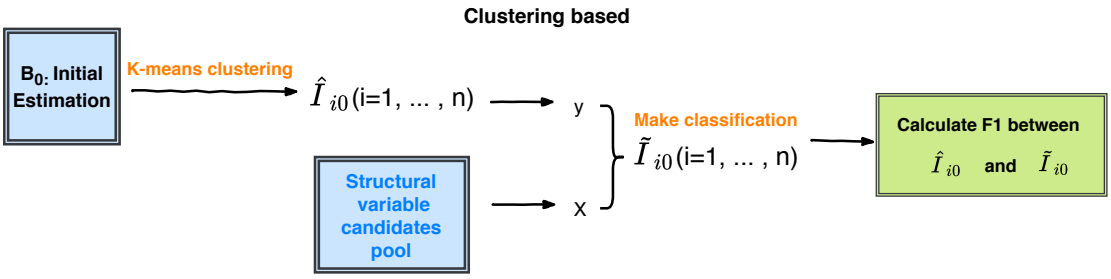


Figure 3.1: Diagram of Clustering Based Narrowing Down Process.

variable as predictors. If the structural variable candidate is proper, the resulted label generated from the classification method, denote as $\tilde{I}_{i0}$, should be similar to the original pilot estimate: $\hat{I}_{i0}$. We quantify the similarity using f1-score ore NMI normalized mutual information (NMI) value (Strehl and Ghosh 2002) between $\hat{I}_{i0}$ and $\tilde{I}_{i0}$. The higher the score, the better the two labels align with each other. Thus, we can select the structural variables with high f1-scores or NMI values to filter out the promising structural variables from the candidate pool.

Narrowing down (ii). Regression based narrowing down process

In some cases, even the candidate pool is properly specified, we still cannot get meaningful results using the first method. The reason behind it is mainly because of the loss of information when performing k-means clustering on matrix $\boldsymbol{B}_0$. In some cases, if $\boldsymbol{B}_0$ does not have a strong clustered pattern, the resulting label will be highly imbalanced with majority of samples having the same label. It makes the further comparison infeasible. Since in method (i), our comparison is completely based on the binary label, it will not be effective in some cases. Instead, we can directly use the pilot estimation matrix to make comparisons. To be more specific, we can either use $\boldsymbol{B}_0$ or get the first principle component PC_1 from $\boldsymbol{B}_0$. After standardizing the structural variable, we regress $\boldsymbol{B}_0$ or PC_1 on each structural variable. Then the model constructed by structural variables with small sum of squares errors (SSEs) would be the promising structural variable candidates. Figure 3.2 shows the diagram of the regression based narrowing down process.

Once we narrow down the structural variable candidates to a relatively small size, we

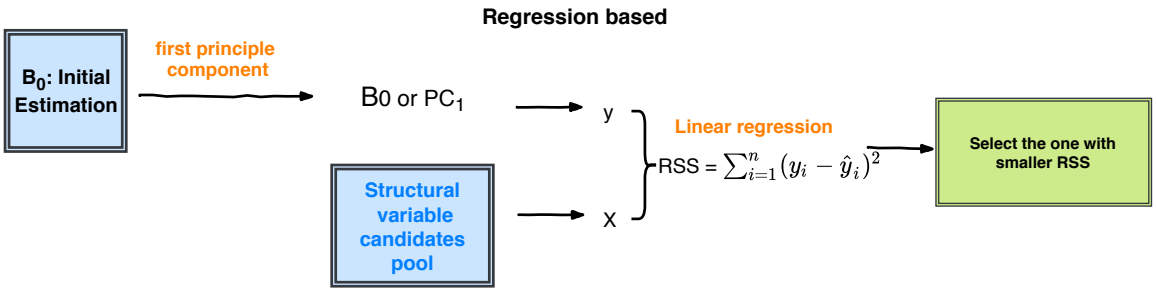


Figure 3.2: Diagram of Regression Based Narrowing Down Process.

can move to the second stage: performing HSA method using the filtered structural variable candidates.

3.3.2 Stage Two: Structural Variable Driven HSA Method

In this stage, we will use HSA method with the help from selected structural variables. The detailed formulation of HSA has been introduced in Chapter 2. We will describe briefly about the corresponding optimization problem, and focus more about the weighting schemes enhanced with the structural variable.

Given a data set with n samples, we want to characterize the relationship between the target variable $\boldsymbol{y} \in \mathbb{R}^n$ and p features $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]^T \in \mathbb{R}^{n \times p}$ using generalized linear models. Let $\boldsymbol{\eta} = [\eta_i] \in \mathbb{R}^n$ $(i = 1, 2, \ldots, n)$ be the systematic component defined as $\eta_i = \boldsymbol{x}_i^T \boldsymbol{\beta}_i$. The loss function of HSA method is:

$$\begin{aligned} f(\boldsymbol{B} \mid \boldsymbol{y}, \boldsymbol{X}) &:= l(\boldsymbol{B} \mid \boldsymbol{y}, \boldsymbol{X}) + P(\boldsymbol{B}) \\ &= -\langle \boldsymbol{y}, \boldsymbol{\eta} \rangle_2 + \langle 1, \tau(\boldsymbol{\eta}) \rangle_2 + P(\boldsymbol{B}), \quad s.t.\ \eta_i = \langle \boldsymbol{X}_i, \boldsymbol{B} \rangle_F, \end{aligned} \tag{3.5}$$

where $l(\boldsymbol{B} \mid \boldsymbol{y}, \boldsymbol{X})$ is the GLM likelihood function, $P(\boldsymbol{B})$ is a regularization term to enforce equi-sparsity. $P(\boldsymbol{B})$ has the following form

$$P(\boldsymbol{B}) = \lambda \sum_{1 \le i < j \le n} w_{ij} \|\boldsymbol{\beta}_i - \boldsymbol{\beta}_j\|_2, \tag{3.6}$$

where w_{ij} is the sample dependent weight to control the grouping pattern, and λ is an overall regularization parameter. In the previous chapter, we introduce two weighting schemes to generate w_{ij}. Now we have structural variables, we can use them to enhance our weighting scheme. To be more specific, we will define a weighting scheme based on the closeness between structural variable pairs.

Since each observation is assigned with a set of independent coefficient, the grouping status of each sample is determined by the closeness of their coefficients. On the other hand, the value of $\boldsymbol{B}$s is hugely affected by the pairwise weights $w_{ij}s$ in the penalty term. As a result, choosing a proper weight is crucial to the success of HSA method. Under the assumption of J-state GLM, the m-dimensional structural variable $\boldsymbol{Z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n)^T \in \mathbb{R}^{n\times m}$ in equation (3.2) directly reflects the closeness between each sample. It indicates that for any sample $(\boldsymbol{x}_i, y_i)$ and sample $(\boldsymbol{x}_j, y_j)$ with $1 \leq i < j \leq n$, if their structural variable $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ are close to each other, then these two samples are more likely to belong to the same group.

Since HSA method has some similarity with convex clustering method in Chen et al. (2015b), we explored their weighting schemes. They proposed a sparse weighting scheme, and the sparsity can expedite the convergence of their algorithm. But it is worthy to note that our weighting scheme cannot be sparse. The main reason is that their method does not involve nested iterations, and the setting is much simpler. On the contrary, the sparsity will make the nested loop converge very slow because of the weak power to enforce rows of $\boldsymbol{B}$ to coalesce. The penalty term would become "sticky", and method will become very sensitive to the regularization parameter λ. Thus, we would not use sparse weight in our method. Instead, we will seek a proper closeness or distance measure to quantify the similarity.

Denote the closeness/distance measure as r_{ij}s. Then the weight w_{ij} should be inversely proportional to it. Since the absolute difference between each sample's structural variable often varies at different scales, it will lead to misleading results sometimes. For example,

Figure 3.3 shows a two dimensional structural variable constructed by $Z = \{z_1, z_2\}$. The decision boundary is a circle and is marked using dotted line. It can be seen that although point A and point B are very close, they belong to different groups. If we use their actual distance, their distance would be quite small, resulting a large weight which will be even larger than ones obtained from (A, C) or (A, D). But A, C and D essentially belong to the same group. As a result, the actual distance leads to some extremely large weight such that it will force the corresponding coefficient vector to be exactly the same. In this way, the grouping status for this pair of samples will be completely determined by their weight.

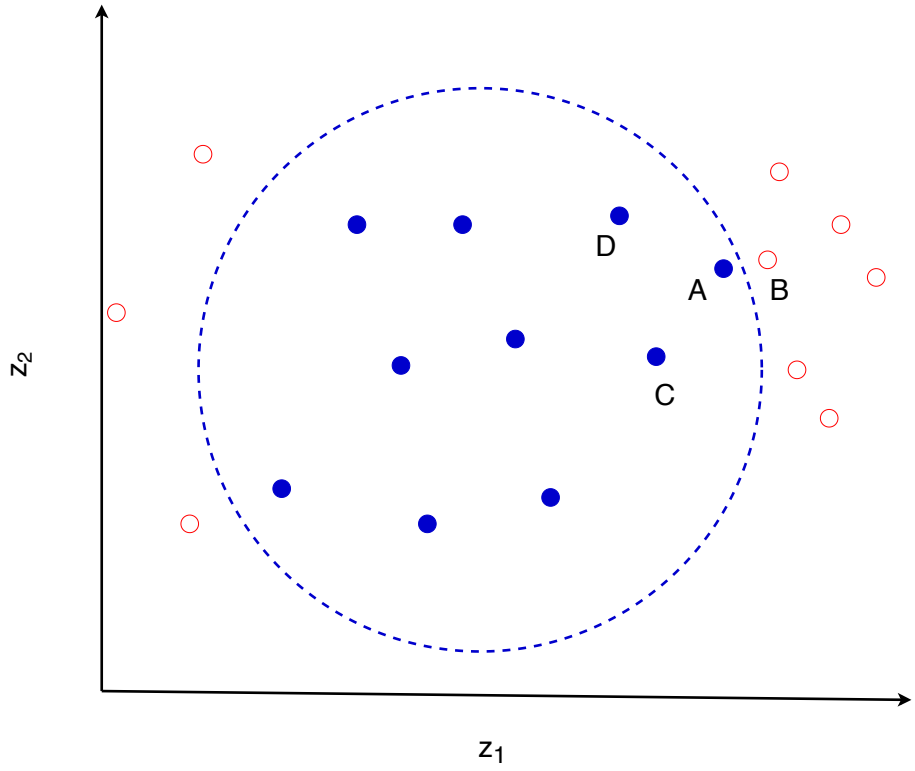


Figure 3.3: Scatter Plot of z.

Instead of the actual distance, we consider to assign a rank distance to each sample pair. For a one-dimensional structural variable $\boldsymbol{z} \in \mathbb{R}^n$, assigning rank is easy. We can get the rank order of $\boldsymbol{z}$ from small to large as $\boldsymbol{r} = (r_1, \ldots, r_n)^T$. Then the weight can be formulated

as

$$\tilde{w}_{ij} = \frac{1}{|r_i - r_j|^k} \tag{3.7}$$

where k is a positive constant to control the strength of difference between each row pair. We can either tune k or set a default value in practice. In the end, we rescale all $\{\tilde{w}_{ij}\}_{1\le i<j\le n}$ to the sum of 1 to get the final weight w_{ij}.

When the structural variable has more than one dimension, assigning a meaningful rank is not so trivial anymore. Consider an m-dimension structural variables $Z = (z_1, \ldots, z_n)^T \in \mathbb{R}^{n\times m}$ $(m > 1)$, for any z_i and z_j $(1 \le i < j \le n)$, define two distance sets: $S_1 = \{\|z_i - z_l\|_2 |\ l = 1, \ldots, n, l \ne i\}$ and $S_2 = \{\|z_j - z_l\|_2 |\ l = 1, \ldots, n, l \ne j\}$. Let r^*_{ij} be the rank of $\|z_i - z_j\|_2$ in the S_1, r^*_{ji} be the rank of $\|z_j - z_i\|_2$ in the S_2. Then we define

$$r_{ij} = \frac{r^*_{ij} + r^*_{ji}}{2} \tag{3.8}$$

as the rank of $\|z_i - z_j\|_2$ among all distances related to z_i and z_j. Let

$$\tilde{w}_{ij} = \frac{1}{r^k_{ij}}, \tag{3.9}$$

where $k > 0$ is the power to control the strength of difference between each row. The larger k is, the intenser the difference of w_{ij} gets. We can either tune k or set a default value like 2 in practice as well. In the end, we rescale $\{\tilde{w}_{ij}\}_{1\le i<j\le n}$ to the sum of 1 to get the final weight w_{ij}.

The ranking distance properly incorporates the information from the structural variable into HSA method. Once we get the state estimate from this stage, we can select out the best model in the next stage. We will talk about how to quantify the quality of the state estimates in the next section.

3.3.3 State Three: Model Selection and Recovering the Decision Boundary

After conducting HSA using different structural variables, we get different state estimates from those structural variables. We need to select out the best one among all candidates.

AIC (Akaike 1998) and BIC (Bhat and Kumar 2010) are both model assessment and selection criteria which are applicable in settings where the fitting is carried out by maximization of the likelihood. Although they are motivated in quite a different way, their forms have a lot of similarities except for the fact that BIC tends to penalize complex models more heavily, giving preference to simpler models in selection. The generic form of AIC and BIC are

$$\begin{aligned} AIC &= -\frac{2}{n} \cdot \text{loglik} + 2 \cdot \frac{d}{n} \\ BIC &= -2 \cdot \text{loglik} + (\log n) \cdot d, \end{aligned}$$

where "loglik" is the maximum log-likelihood value, d is the number of parameters in the model. For a k-state linear regression model, we get

$$AIC(\boldsymbol{\beta}, \sigma^2) = -\frac{2}{n} \sum_{i=1}^{n} \log(C_i) + 2 \cdot \frac{d}{n} \tag{3.10}$$

$$BIC(\boldsymbol{\beta}, \sigma^2) = -2 \sum_{i=1}^{n} \log(C_i) + d \log(n), \tag{3.11}$$

where

$$C_i = \frac{1}{\sqrt{2\pi}\hat{\sigma}} \exp\left(-\frac{(y_i - \boldsymbol{X}_i \hat{\boldsymbol{\beta}}^{(\hat{I}_i)})^2}{2\hat{\sigma}^2}\right).$$

Here $\hat{\sigma}$ is the maximum likelihood estimation (MLE) based on all samples in the data, and $\hat{\boldsymbol{\beta}}^{(\hat{I}_i)}$ is the corresponding MLE using HSA estimated state $\hat{I}_i \in \{1, \ldots, J\}$. d is the number of parameters in the model. We can select out the best model with smaller AIC or BIC values.

Once we get the state estimates of each sample, we can use them to recover the decision boundary with respect to the selected structural variable. There are lots of classification algorithms we can use, like logistic regression, support vector machine, etc.. We can choose an appropriate method based on real needs and obtain the final decision boundary of the structural variable. For example, if the estimated states are binary ($\hat{I}_i \in \{1, 2\}$), we can use linear support vector machine (linear SVM) to construct linear decision boundary using the value pair $(\hat{I}_i, \boldsymbol{z}_i)$. The corresponding optimization problem is:

$$\begin{aligned} &\max_{\boldsymbol{\gamma}, \gamma_0, \|\boldsymbol{\gamma}\|_2 = 1} M, \\ &\text{s.t. } (2\hat{I}_i - 3)(\boldsymbol{z}_i^T \boldsymbol{\gamma} + \gamma_0) \geq M, \quad i = 1, \ldots, n. \end{aligned}$$

The corresponding decision boundary would be $\{\boldsymbol{z} | \boldsymbol{z}^T \boldsymbol{\gamma} + \gamma_0 = 0\}$. Similarly, we can also use other classification methods to recover the decision boundary in real practice.

3.4 Simulation Experiments

In this section, we conduct a series of simulation experiments to explore the performance of SD-HSA modeling framework. Both linear regression and logistic regression will be used to test the framework.

Example 1:

We first consider a 2-state linear regression with sample size $n = 150$.

$$y_i = \begin{cases} -0.5 - 0.1x_{1i} + 0.4x_{2i} + \epsilon_i & I_i = 1, \\ 0.2 + 0.5x_{1i} - 0.1x_{2i} + \epsilon_i & I_i = 2. \end{cases}$$

where $\epsilon_i \sim N(0, 0.3^2)$ $(i = 1, \ldots, n)$. The 3D-scatter plot of y v.s. X is shown in Figure 3.4.

Assume the true structural variable space has a linear boundary as shown in Figure 3.5. The horizontal distances d between the two groups are set as 0, 0.1 and 0.2 respectively.

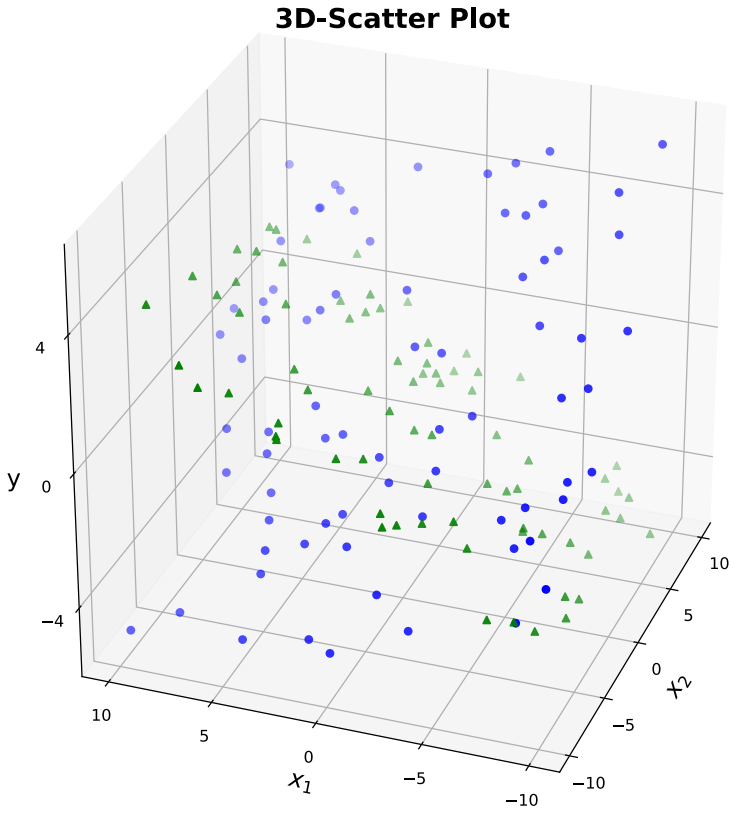


Figure 3.4: Example 1 - 3D Scatter Plot of y v.s. X.
Two groups are marked with green triangular dots and blue dots. It can be seen that these two groups have a lot of overlaps within the 3-D space.

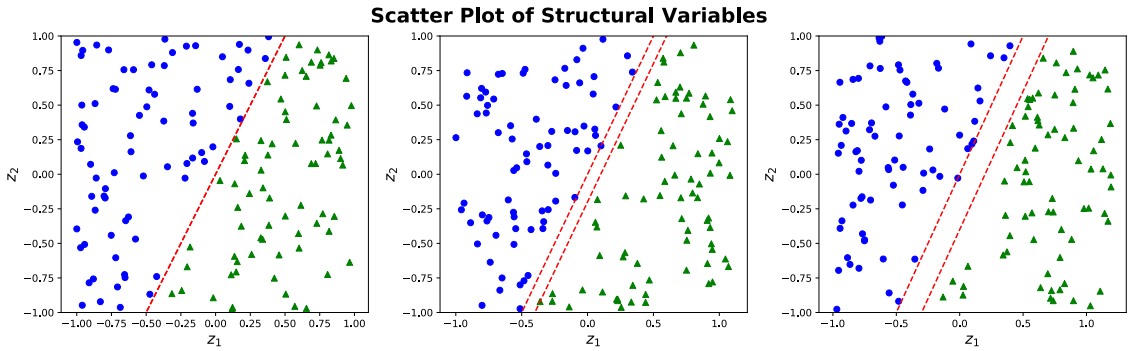


Figure 3.5: Example 1 - Scatter Plots of Structural Variables.
As d increases, the two groups plotted in different colors are more separable.

With smaller d, these two groups will be more difficult to separate. We repeat each setting for 50 times.

For the structural variable candidate pool, we collect all possible combinations of z_1, z_2, z_1^2, z_2^2, $z_1 z_2$, which results in 31 structural variables in total.

At the first stage, in order to get the pilot estimation of $\boldsymbol{\beta}_i$ $(i = 1, \ldots, n)$, we use the "initial HSA method" described in Section 3.3.1. Then we apply the clustering based narrowing down process on 31 structural variables to calculate the F1-score. We selected out the top 3 candidates from the 31 variables. Table 3.1 shows the percentage of experiments among the 50 experiments which successfully included (z_1, z_2) within the top 3 selected candidates based on their F1-scores. All of the success rates from different ds are quite high. This indicates that the narrowing down process is effective.

Success Rates			
d	$d = 0$	$d = 0.2$	$d = 0.2$
Percentage	94%	100%	96%

Table 3.1: Example 1 - Success Rates of Top 3 Selected Candidates.

Next, we use HSA method to get the state estimates using the top 3 selected structural variables. For each experiment, we also selected out the best model using BIC values. Same as Chapter 2, we use the rand index (Rand 1971b) to measure the performance of HSA method. Figure 3.6 shows the rand indices between the grouping results and the true

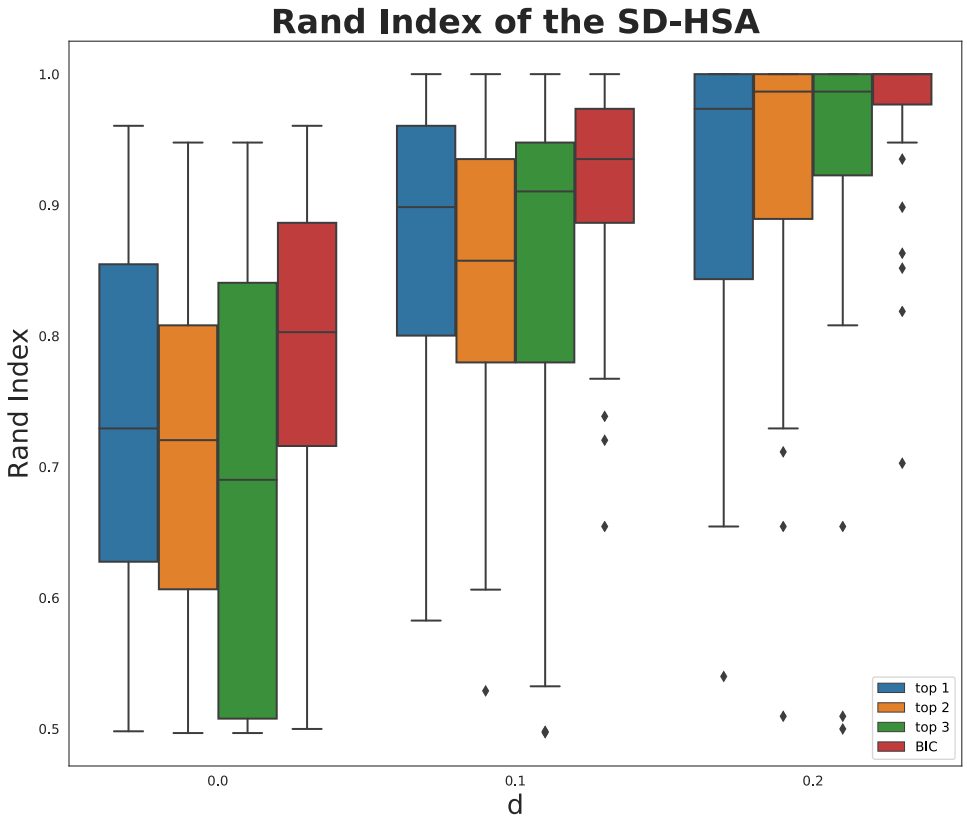


Figure 3.6: Example 1 - Rand Indices of the top 3 Structural Variables.

state of each sample. The grouping results were generated from top 3 structural variables and the best one selected using BIC values correspondingly. It can be seen that as distance between the two groups gets larger, the rand indices also get higher, and the BIC selected model consistently has the highest rand index among all of them. Thus, we successfully auto-grouped the data in this example, and selected out the best model using BIC values.

We also plot the decision boundary under three different distances corresponding to their median rand indices. Figure 3.7 shows the results. It can be seen that as d gets larger, the decision boundary separates the two groups better. This also confirms with out intuition.

Example 2:

We consider a logistic regression model. Consider a 2-state logistic regression with sample

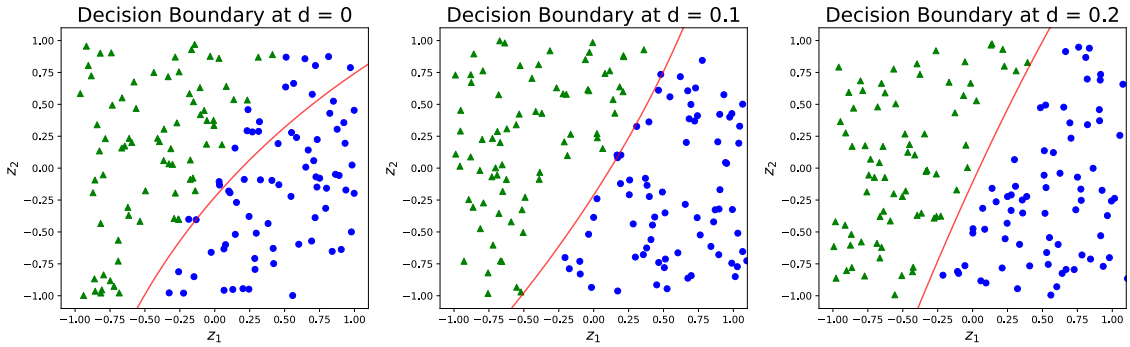


Figure 3.7: Example 1 - Decision Boundary of the Structural Variable Space.

size $n = 150$.

$$logit(p_i) = \log \frac{p_i}{1-p_i} = \begin{cases} -0.5 - 0.1x_{1i} + 0.4x_{2i} & I_i = 1, \\ 0.2 + 0.5x_{1i} - 0.1x_{2i} & I_i = 2. \end{cases}$$

The 3D-scatter plot of y v.s. X is shown in Figure 3.8. Similar as before, we also assume that the structural candidate pool is constructed with all possible combinations of z_1, z_2, z_1^2, z_2^2, $z_1 z_2$, which results in 31 different structural variables as shown in table **??**. The true decision boundary is shown in Figure 3.9. The horizontal distances d between the group drawn in blue and green are set as 0, 0.2 and 0.4 respectively. We also repeat the experiment for 50 times under the each scenario.

At the first stage, we use the "Bayesian method" in Section 3.3.1 to get the pilot estimation $\boldsymbol{\beta}_i$, and narrow down the potential candidate structural variables using the clustering based method. Table 3.2 shows the percentage of experiments among the 50 experiments which has been successfully included (z_1, z_2) within the top 3 selected candidates based on their F1-scores. The success rate is a little bit lower than the cases in example 1, but still quite high overall. This indicates that although grouping for logistic regression seems more difficult than linear regression because of the binary response variable giving limited information to use, we still use out method to effectively narrow down potential structural variables.

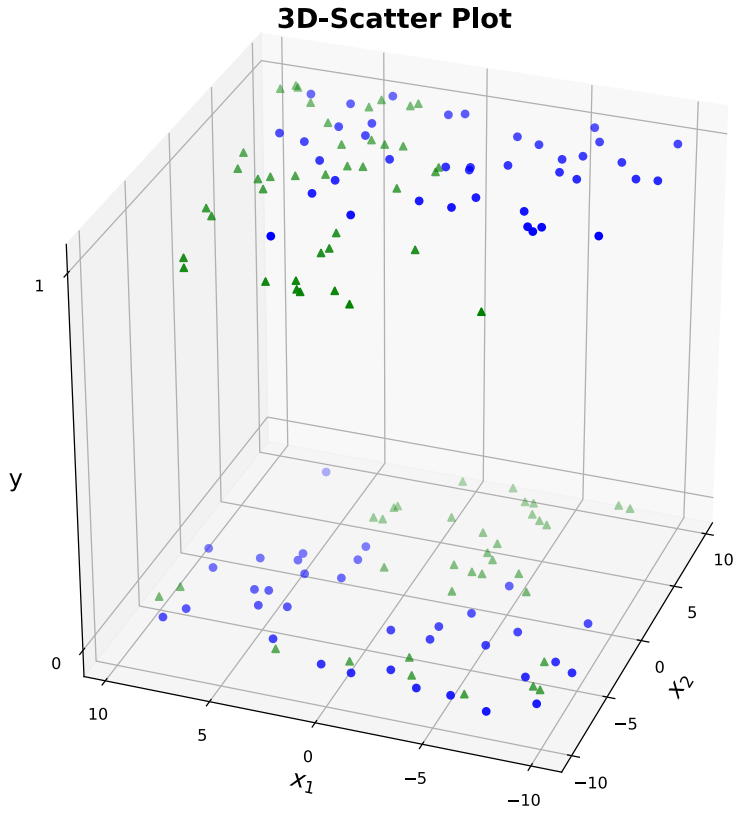


Figure 3.8: 3D-scatter Plot of $\boldsymbol{y}$ v.s. $\boldsymbol{X}$.
$\boldsymbol{y}$ is binary, and there exist samples of both groups within each category of $\boldsymbol{y}$. In another word, the two groups are overlapped with each other. The average sample variance is around 0.1.

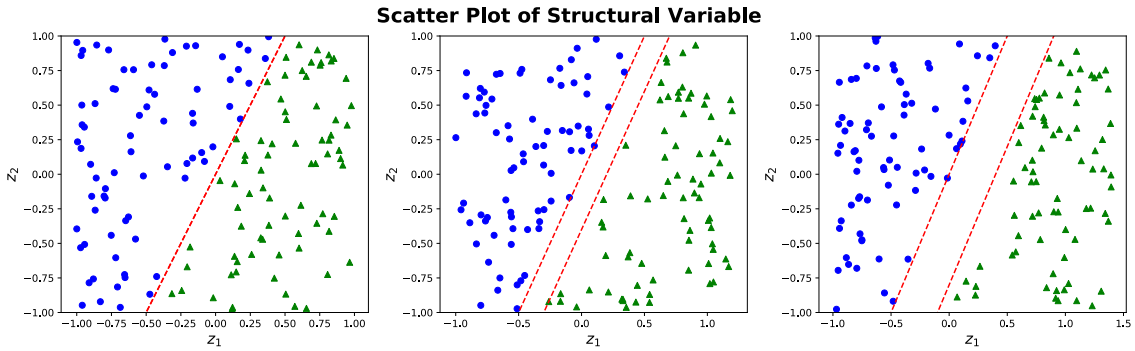


Figure 3.9: Example 2 - Scatter Plots of Structural Variables.

Success Rates			
d	$d = 0$	$d = 0.2$	$d = 0.4$
Percentage	88%	88%	94%

Table 3.2: Example 2 - Success Rates of Top 3 Selected Candidates.

Same as before, We adopt HSA method using the top 3 selected structural variables. For each experiment, we also selected out the best model using BIC values and calculated the corresponding rand index. Figure 3.10 shows the rand indices for the top 3 structural variables and the best one selected using BIC values. We can get similar conclusion as in example 1. The models selected using BIC values also have the smallest variance in terms of rand indices. Finally, we plotted the decision boundary under the three different distances corresponding to their median rand indices. Figure 3.11 shows the results. It can be seen that as d gets larger, the decision boundary separates the two groups better.

Example 3:

We consider a linear regression with a circular structural decision boundary. The two-state linear regression has the same form as in equation (3.4). Assume the true structural variable space has a circular boundary with different distances as shown in Figure 3.12. The distances d between the two groups are set as 0, 0.1 and 0.2 respectively. We repeat the experiment for 50 times under the each scenario.

At the first stage, we used the "initial HSA method" to get pilot estimates $\boldsymbol{\beta}_{i0}$ and narrow down the potential candidate structural variables through clustering based method. Table

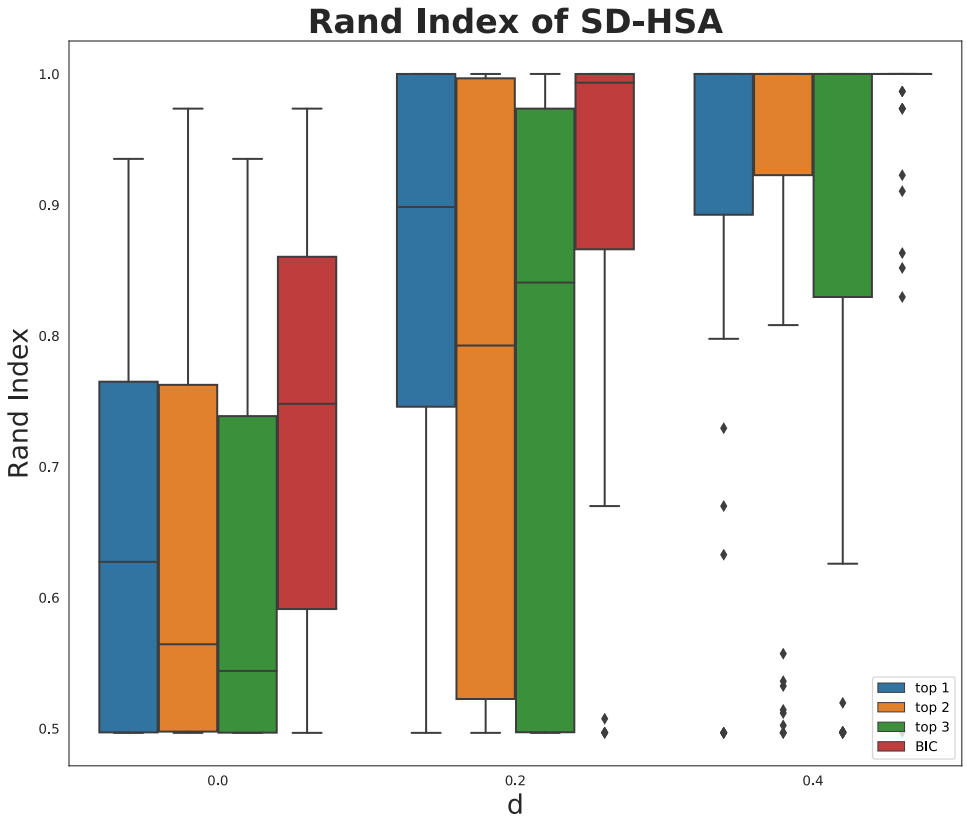


Figure 3.10: Example 2 - Rand Indices of the Top 3 Structural Variables.

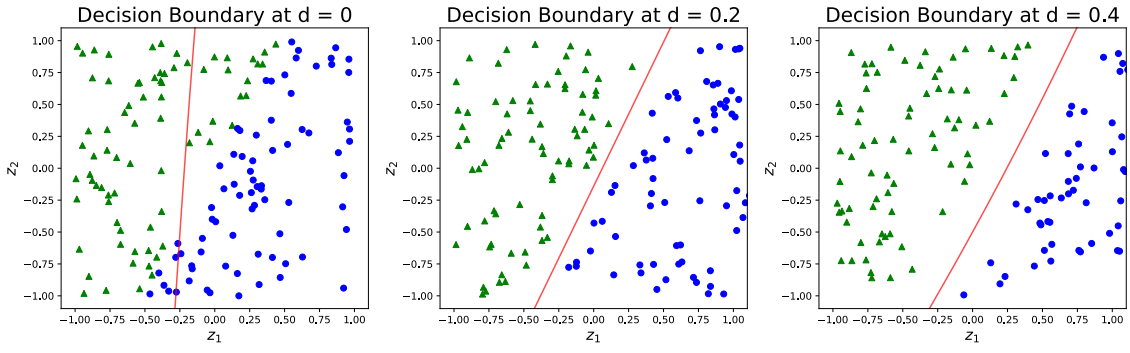


Figure 3.11: Example 2 - Decision Boundary of the Structural Variable Space

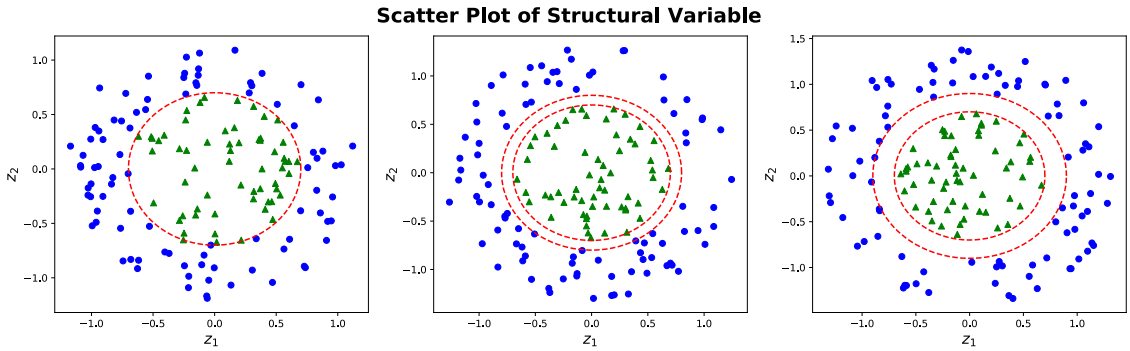


Figure 3.12: Example 3 - Scatter Plots of Structural Variables.
The two groups are more separable as d gets larger.

3.3 shows the percentage of experiments among the 50 experiments which successfully included (z_1^2, z_2^2) within the top 3 selected candidates based on their F1-scores. It can be seen that the success rates are also very high overall.

Success Rates			
d	$d = 0$	$d = 0.1$	$d = 0.2$
Percentage	82%	90%	92%

Table 3.3: Example 3 - Success Rates of Top 3 Selected Candidates

Next, we conducted HSA process using the top 3 selected structural variables. Same as before, we also selected out the best model using BIC values and calculated the corresponding rand index. Figure 3.13 shows the rand indices for the top 3 structural variables and the best one selected using BIC values. We also got similar conclusions as previous examples.

Finally, we plotted the decision boundary under the three different distances corresponding to their median rand indices. Figure 3.14 shows the results. It can be seen that the decision boundaries are recovered quite well for the three cases.

3.5 Real Data Examples

In this section, we use SD-GLM to analyze U.S. economic indicators: nonfarm payroll numbers and the unemployment rate. The data can be found from Bureau of Labor Statis-

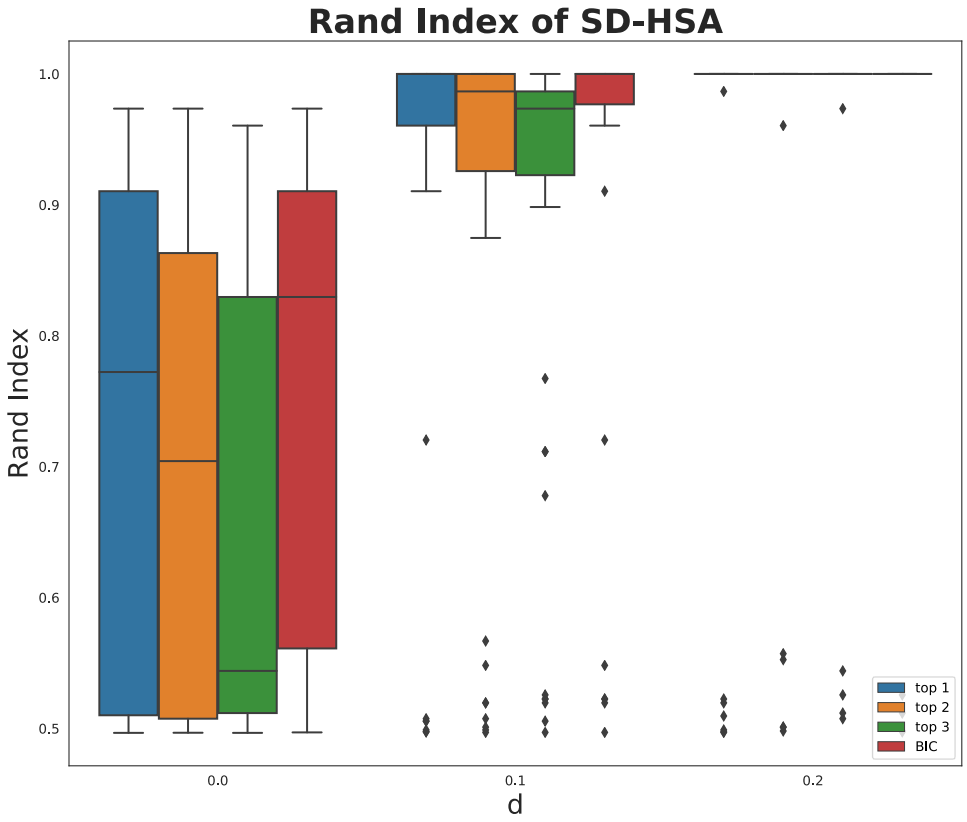


Figure 3.13: Example 3 - Rand Indices of the Top 3 Structural Variables.

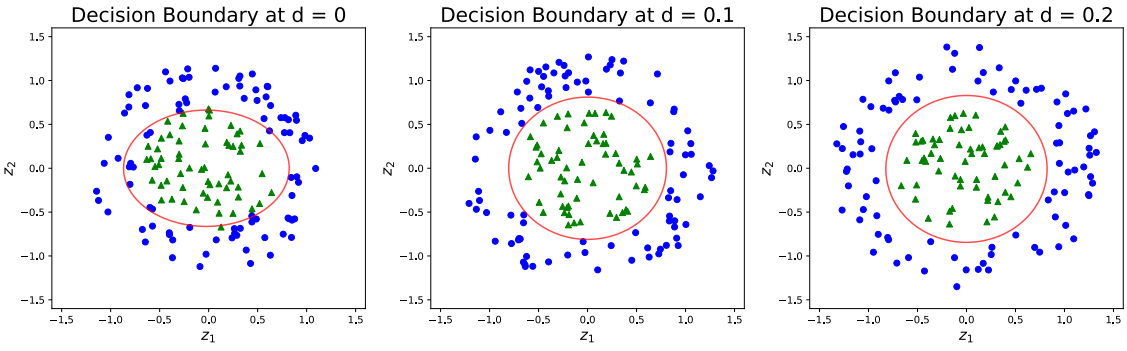


Figure 3.14: Example 3 - Decision Boundary of the Structural Variable Space

tics (www.bls.gov). The two data sets have been analyzed in Wu and Chen (2007). We will make comparison with the model obtained from their analysis. The first series ranges from January, 1939 to March, 2004, the second from January, 1948 to March, 2004. Both are seasonally adjusted.

3.5.1 U.S. Nonfarm Payroll Numbers

We make the similar pre-processing procedure to the series as in Wu and Chen (2007). We transform the original monthly data to quarterly difference

$$Q_t = \frac{P_{3(t-1)} + P_{3(t-2)} + P_{3(t-3)}}{3}, \quad \text{and} \quad Y_t = \frac{Q_t - Q_{t-1}}{5,000} \tag{3.12}$$

for $t = 1, \ldots, 260$. Here P_t is the monthly payroll number, Q_t represents the quarterly average, and Y_t represents the quarterly difference. We let the unit of Y_t be $5,000$.

An AR(2) model can be fitted with $Y_t = \phi_0 + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \epsilon_t$. The MLE estimates are $\hat{\phi}_0 = 0.2233$, $\hat{\phi}_1 = 1.0054$ and $\hat{\phi}_2 = -0.3100$. The total series length is $T = 250$. It can be verified that the estimated model is stationary.

We can build an 2-state AR(2) model on Y_t. Given some structural variable defined on a m-dimensional space $\boldsymbol{z}_t = (z_1, \ldots, z_m)^T \in \mathbb{R}^{m\times 1}$, a 2-state AR(2) model can be defined as below:

$$Y_t = \begin{cases} \phi_0^{(1)} + \phi_1^{(1)} Y_{t-1} + \phi_2^{(1)} Y_{t-2} + \epsilon_1 & I_t = 1, \\ \phi_0^{(2)} + \phi_1^{(2)} Y_{t-1} + \phi_2^{(2)} Y_{t-2} + \epsilon_2 & I_t = 2, \end{cases} \tag{3.13}$$

$$I_t = k, \quad \text{if } \boldsymbol{z}_t \in R_k \ (k = 1, 2) \tag{3.14}$$

where R_k is the decision region determined by the structural variable. The model in each state is equivalent to a linear regression with predictors Y_{t-1} and Y_{t-2}.

Using the same setting in Wu and Chen (2007), assuming the structural variable candidate pool is constructed by series up to eight lag variable and its squares. We consider linear

combination up to three variables in the candidate pool. It results in totally 696 possible candidates. We first get the pilot estimation $\boldsymbol{\beta}_{i0}$ $(i = 1, \ldots, 250)$ using Bayesian method described in Section 3.3.1. It formulates the pilot estimation matrix $\boldsymbol{B}_0$ (with the intercept removed). Then we get the first principle component $\boldsymbol{p}_1$. We treat it as our response variable ($\boldsymbol{y}$), and all the potential structural variable candidates as different features ($\boldsymbol{X}$s) and fit different linear regressions. Then we found out the ones with smaller residual of sum of squares. We choose the top 4, 52, 52 among one, two, three-variable combinations within the candidate pool, and narrow down the total candidates from 696 to 108 variables in total.

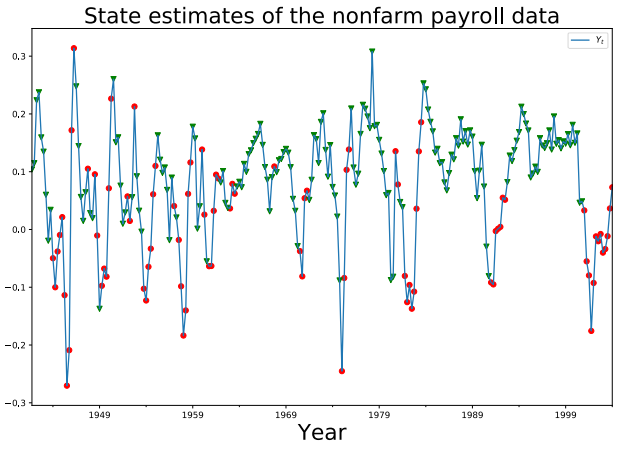
Next, we conduct HSA process on the selected 108 variables. The model with smallest BIC is derived from the structural variable Y_{t-2}. As a result, we use Y_{t-2} as our best structural variable and recover the decision boundary using linearSVM in Python Scikit-learn package. The left penal of Figure 3.15 shows the final estimated states for Y_t. It can be seen that samples in the group in green dots seems to be consecutive and very close to each other. It might correspond to the period when the quarterly difference does not vary a lot. The right penal of Figure 3.15 show the decision boundary constructed by Y_{t-2}. The decision boundary cuts the series nearly in the middle.

We also compared our model with the best model from Wu and Chen (2007). Given the state estimates $\hat{I}_t$ $(t = 1, \ldots, T)$ of each sample, we divide the data into 2 groups. For each group, we can get its MLE ($\hat{\phi}^{(\hat{I}_t)}$) and calculate the sum of the square of errors (SSE). The *hardSSE* of the data is defined as the sum of SSE of the two groups. Namely

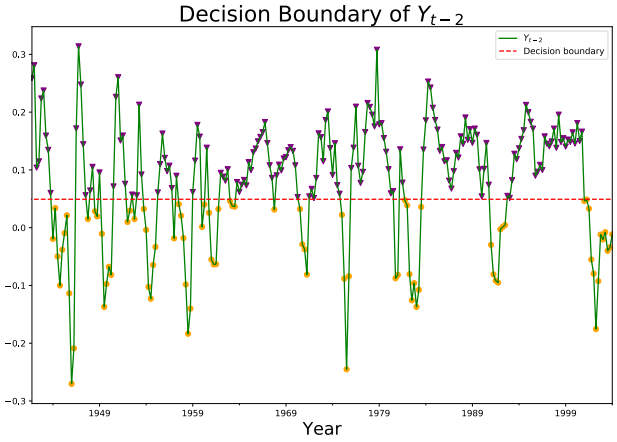
$$\text{Hard SSE: } \sum_{t=p}^{T} (Y_t - \boldsymbol{y}_{t-1} \hat{\phi}^{(\hat{I}_t)})^2 \tag{3.15}$$

where $p = 2$ in our data. Table 3.4 provides the hard SSE values of the best structural variable shown in Wu and Chen (2007) and the best model in our method. It can be seen that our model has a smaller hardSSE value.

In summary, the final two-state AR(2) model has the following form. The decision



(a) State Estimates for Y_t.
The green triangular dots and red dots stand for two different groups.



(b) Decision Boundary for Y_{t-2}.
The decision boundary degenerates to a threshold in this case.

Figure 3.15: Results of Nonfarm Data.

Table 3.4: Model comparison for U.S. Nonfarm Payroll Data.

Model	**Threshold/Structural Var.**	**Hard SSE**
TD-SAR(2)	(Y_{t-2}, Y_{t-3}^2)	81.82
SD-HSA	(Y_{t-2})	80.4806

boundary is constructed by Y_{t-2}. The two-state AR(2) model is

$$Y_t = \begin{cases} -0.0104 + 0.0980Y_{t-1} - 0.0792Y_{t-2} + \epsilon_1 & I_t = 1, \\ 0.0237 + 0.0896Y_{t-1} - 0.0174Y_{t-2} + \epsilon_2 & I_t = 2, \end{cases}$$

$$R_1 = \{Y_{t-2} | Y_{t-2} \leq 0.0491\},$$

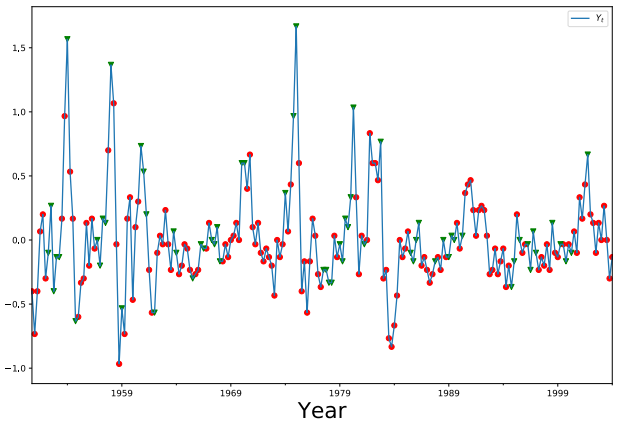
$$R_2 = \{Y_{t-2} | Y_{t-2} > 0.0491\}.$$

3.5.2 U.S. Unemployment Rate

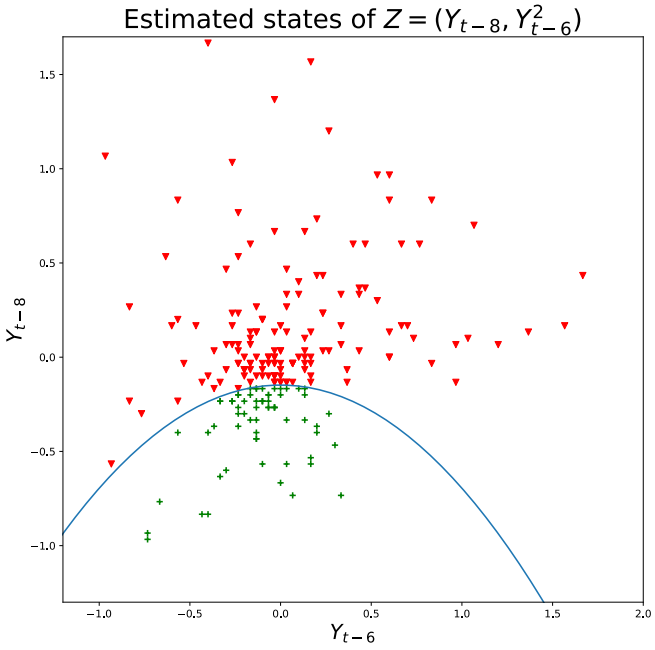
We obtain the quarterly differences on the U.S. unemployment rate data (as with the non-farm payroll series). The total length of the series is 214. We also use SD-HSA to get a 2-state AR(2) with different structural variables. The structural variable candidate pool includes one-variable and two-variable combinations from lag 1 to lag 8 of the observed series and their squares. This results in 136 candidates in total.

At the first stage of narrowing down the potential structural variables, we use the regression based narrowing down process similar as in the previous example. We select out the top 36 structural variables from the candidate pool and use them to conduct HSA process. Based on 36 different estimated states from structural variable, we select the one with the smallest BIC value, and the corresponding structural variable is (Y_{t-8}, Y_{t-6}^2). The left penal of Figure 3.16 shows the final estimated states for Y_t. It can be seen that samples in the group in green dots mainly correspond to the upper change points. The right penal of Figure 3.16 show the decision boundary constructed by (Y_{t-8}, Y_{t-6}^2) using linear SVM.

Table 3.5 shows the performance comparison of the best two models using our model in



(a) State Estimates for Y_t.
The green triangular dots and red dots stand for two different groups.



(b) Decision Boundary for (Y_{t-8}, Y_{t-6}).
The decision boundary is quadratic with respect to Y_{t-6}.

Figure 3.16: Results of Unemployment Data.

two-variable combination in terms of the values of HardSSE. The values are smaller than the best model selected from Wu and Chen (2007).

Table 3.5: Model Comparison for U.S. Unemployment Rate Data.

Model	**Threshold/Structural Var.**	**Hard SSE**
TD-SAR(2)	(Y_{t-1}, Y_{t-2})	16.84
SD-HSA	(Y_{t-8}, Y_{t-6}^2)	15.9966

The final 2-state AR(2) model has the following form. The structural variable is $\boldsymbol{Z} = (\boldsymbol{y}_{t-8}, \boldsymbol{y}_{t-6}^2)$. The two-state AR(2) model is

$$
\begin{aligned}
Y_t &= \begin{cases} -0.0322 + 0.5403Y_{t-1} - 0.1255Y_{t-2} + \epsilon_1 & I_t = 1, \\ 0.0906 + 1.0313Y_{t-1} + 0.1190Y_{t-2} + \epsilon_2 & I_t = 2, \end{cases} \\
R_1 &= \{(Y_{t-8}, Y_{t-6}^2) | (Y_{t-8} + 0.5475Y_{t-6}^2 + 0.1478 > 0\}, \\
R_2 &= \{(Y_{t-8}, Y_{t-6}^2) | (Y_{t-8} + 0.5475Y_{t-6}^2 + 0.1478 \leq 0\}.
\end{aligned}
$$

3.6 Conclusion

In this chapter, we propose a three-stage modeling framework based on HSA method. The SD-HSA framework enhanced the performance of the HSA by incorporating the information of structural variables into HSA weighting scheme. We narrow down the potential candidates of structural variables to reduce the computational cost at the first stage. At the second stage, we conduct HSA method within a smaller number of structural variable candidates to get the state estimates of each sample. At last, we choose the best model and structural variable by model selection criteria like AIC or BIC values. In the end, we can recover the decision boundary based on the estimated states using classification methods.

Through the simulation examples, we can see that the overall performance of HSA method improved a lot compared with similar settings without any structural variables, and

we successfully recovered some non-linear decision boundaries of the structural variable as well. In summary, the framework provides us with more potentiality of applications for HSA method. We also apply the modeling framework to the U.S. nonfarm and unemployment data, and compare the performance with the TD-SAR proposed in Wu and Chen (2007) in terms of hard SSE values. We get smaller hard SSE values for both two cases.

In summary, Chapter 2 and Chapter 3 mainly talk about two different application scenarios for HSA method. When there is no structural variable, we can use HSA method to explore the hidden heterogeneity. When structural variables exist, we can use SD-HSA modeling framework to recover the decision boundary determined by the structural variable. In this way, we can build better model with the help of HSA.

CHAPTER 4
LOCO (LOCAL CONNECTIVITY) SCORE: AN INTERPRETABLE METHOD FOR DETECTING LOCAL OUTLIERS

4.1 Introduction

Outlier detection is considered as a critical task in many real applications, such as fraud detection, healthcare monitoring and industrial damage detection, etc.. Detecting outliers from a pattern is a popular problem. Generally, outlier can be defined as an observation that deviates a lot from other observations as to arouse suspicion that it was generated by a different mechanism (Breunig et al. 2000). The outliers originally existed as the by-product of many clustering algorithms. Thus, it is mainly based on the entire data set, and the early outlier detection methods obtained a set of global outliers. However, with the rapid development of information technology, the structure of data sources is becoming more and more complex. Sometimes, due to the instability of data collection and transmission technology, etc., the data sets obtained are often incomplete in terms of time and space. In this scenario, we only care about the change of things in a local scope. The corresponding outliers obtained would be local outliers. In some other cases, a data point is considered as a contextual outlier if its value significantly deviates from the rest of the data points in the same context. For example, in the anomaly detection of vessels in maritime transportation system, trajectories of different vessels might overlap with each other. We would hardly find any global outliers from a bunch of overlapped trajectories, but some vessels might have abnormal behavior like hanging around in a small region or extreme high speed compared with their neighborhood vessels. Thus, we also need local outlier detection techniques in this scenario. In this chapter, we propose a local outlier detection method: LoCO (local COnnectivity) score. It is a directed neighborhood-based approach using similarity scores

of each individual's directed neighborhood information. This method is easy to interpret, and can also be used to calculate a "p-value" using conformal prediction techniques (Xie and Zheng 2020b) to judge each data subject's outlyingness conveniently.

Outlier detection problems have been extensively studied. In general, existing methods can be grouped as follows: (1) clustering based approach (Guha et al. (2003), Cao et al. (2006)). The main goal of such systems is to build clusters. Points that are far away from cluster centroid are declared as outliers. As a result, this techniques find the global outlier. (2) depth based approach (Tukey (1977), Ruts and Rousseeuw (1996), Johnson, Kwok, and Ng (1998)). This type of outlier detection searches for outliers at the border of the data space but independent of statistical distributions. Outliers are subjects on outer layers. (3). Distance based approach (Knorr and Ng (1997), Knox and Ng (1998), Knorr and Ng (1999), Ramaswamy, Rastogi, and Shim (2000), Fan et al. (2006), Ghoting, Parthasarathy, and Otey (2008)). Distance based techniques are not able to detect geterogeneous densisties in data and outliers in heterogeneous densities. (4). Density-based approaches: LOF and its variations (Breunig et al. (2000), Tang et al. (2002), Lazarevic and Kumar (2005), Kriegel et al. (2009a), Kriegel et al. (2011)). LOF quantifies the degree of outlying of a data subject to be the ratio of its density and the average density of its neighboring subjects. However, this method is not very sensitive to the difference of the density distribution of the subject's neighborhood, and its resulting quotient-values are hard to interpret. As a result. there is no clear threshold to tell us when a point is an outlier. In one data set, a value of 1.3 might already be an outlier. In another data set and parametrization with strong local fluctuations, a value of 2 could still be an inlier. Some variations of LOF include local outlier probability (LoOP) (Kriegel et al. 2009a). It interprets and unifies outlier scores (Kriegel et al. 2011) which uses local statistics or statistical scaling with a resulting values scaled to a value range of $[0, 1]$. Some other methods include feature bagging (Lazarevic and Kumar 2005) that runs LOF on multiple projections and combines the results to improve detection quantities in high dimensions. Those variations of LOF origin from a similar idea with LOF

and improve it from different aspects. Different from them, our newly proposed method aims to quantify the degree of outlyingness based on a directed neighborhood network. We will show it not only has good statistical interpretation, but also performs better in some challenging scenarios for LOF.

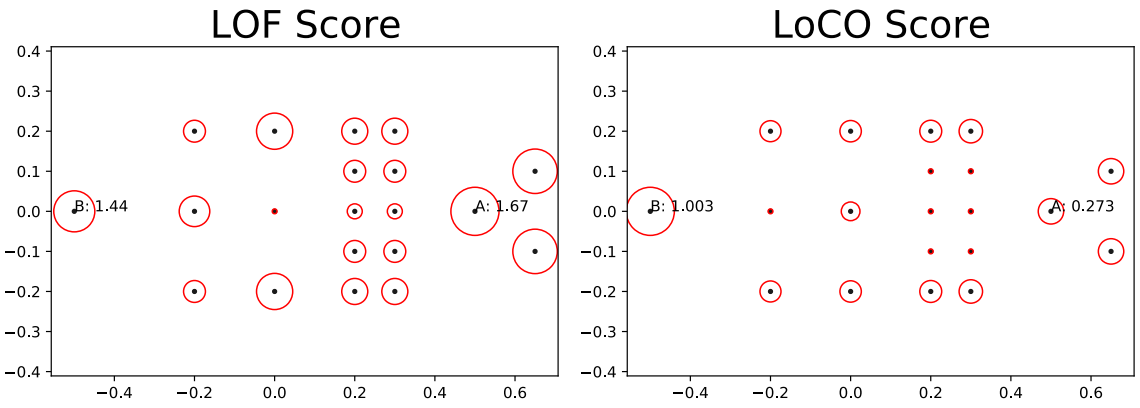
Since we compare our method's performance with the LOF method, it is necessary to illustrate an example to show a case which LOF could not handle well. We first look at a data set with samples shown in Figure 4.1. We calculate the LOF score of each subject when $k = 3$, and represent them as each circle's radius in Figure 4.1a. It can be seen that subject A and subject B have LOF values 1.67 and 1.44 respectively. But it is obvious that B looks more isolated than A. So subject B should have a larger outlying score intuitively. This counter-intuitive result is mainly caused by the density changes within the data. LOF cannot properly handle the data set with density changes mainly because of the fact that it only considers each data subject's neighborhood points independently. But the neighborhood points could not proper reflect the true context of the data subject with changing densities. This motivate us to consider a directed-neighborhood network and calculate similarities based on it. Simulation examples have shown that this method is more robust to density changes.

The rest of the chapter is organized as follows. we first introduce LoCO score and its auxiliary notations. Then we analyze some properties of LoCO score. Furthermore, we calculate the "p-value" of each data subject by incorporating LoCO scores into conformal prediction's framework. At last, we provide numerical and real data examples to explore the performance of the proposed method.

4.2 LoCO Score

4.2.1 Notations and Definitions

Suppose we have a dataset $\mathcal{D}$ with n subjects $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$. The pairwise distance based on some distance or similarity measure between $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$ is denoted as d_{ij} $(1 \leq i \neq j \leq n)$. We



(a) LOF score. (b) LoCO score.

Figure 4.1: Outlying Scores at $k = 3$.
Point B looks absolutely more isolated compared with point A. But because of the density changes, when $k = 3$, LOF generates a higher score for point A. Instead, the outlying score given by LoCO is more reasonable.

introduce the following definitions:

Definition (k**-distance of** $\boldsymbol{s}_i$)

Given any positive integer k, the k-distance of an subject $\boldsymbol{s}_i$, denoted as $k - dist(\boldsymbol{s}_i)$, is defined as the distance $d_{i,j}$ between $\boldsymbol{s}_i$ and $\boldsymbol{s}_j \in \mathcal{D}$ such that

1. for at least k subjects $\boldsymbol{s}_{j'} \in \mathcal{D}\backslash\{\boldsymbol{s}_i\}$ it holds that $d_{i,j'} \leq d_{i,j}$ and

2. for at most $k - 1$ subjects $\boldsymbol{s}_{j'} \in \mathcal{D}\backslash\{\boldsymbol{s}_i\}$ it holds that $d_{i,j'} < d_{i,j}$.

Definition (k**-neighborhood set of** $\boldsymbol{s}_i$) All points $\boldsymbol{s}_{j'} \in \mathcal{D}$ satisfying $d_{i,j'} \leq k - dist(\boldsymbol{s}_i)$ construct the k-nearest neighborhood set of $\boldsymbol{s}_i$. Namely, $\mathcal{N}_k(\boldsymbol{s}_i) := \{s_j \in \mathcal{D}\backslash\{\boldsymbol{s}_i\} \mid d_{i,j} \leq k - dist(\boldsymbol{s}_i)\}$.

Definition (k**-connectivity set of** $\boldsymbol{s}_i$) Given positive integer k, the connectivity set of subject $\boldsymbol{s}_i$, denote as $C_k(\boldsymbol{s}_i)$, is the set constructed by subjects $\boldsymbol{s}_{j'} \in \mathcal{D}$ such that $\boldsymbol{s}_i \in \mathcal{N}_k(\boldsymbol{s}_{j'})$.

It is worthy to note that for any data subject $\boldsymbol{s}_i \in \mathcal{D}$ and positive integer k, $\mathcal{N}_k(\boldsymbol{s}_i)$ will not be empty. But it is possible for $C_k(\boldsymbol{s}_i)$ to be empty. If $C_k(\boldsymbol{s}_i) = \emptyset$, it indicates that $\boldsymbol{s}_i$ is a very isolated point such that no point treats it as its neighborhood. Thus, the degree

of outlyingness of s_i could be very high. On the other hand, as k increases, the difference between the neighborhood set and connectivity set for each data subject will be lessened gradually. As a result, the choice of k will also affect the magnitude of difference between the neighborhood set and connectivity set.

Definition (**Popularity score of s_i**) For any subject $s_i \in \mathcal{D}$, the popularity score of s_i is defined as

$$Pop(s_i) := \frac{|C_k(s_i) \bigcap \mathcal{N}_k(s_i)|}{|\mathcal{N}_k(s_i)|}. \tag{4.1}$$

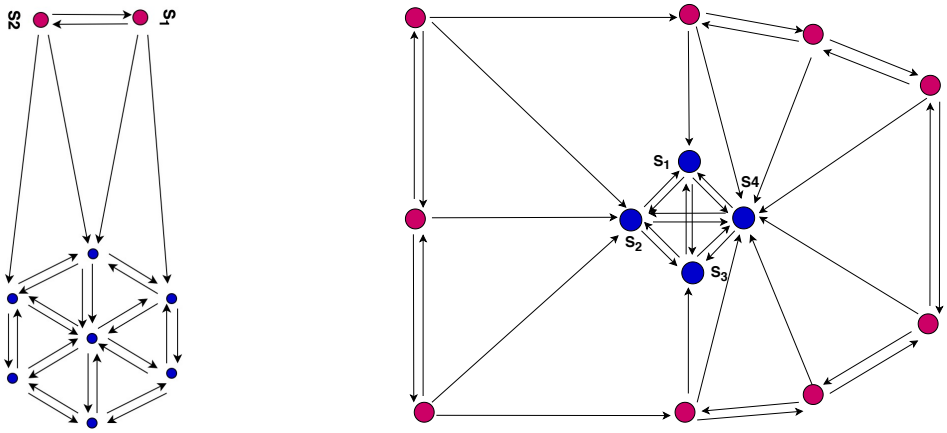
It quantifies the proportion of subjects among the neighborhood of s_i that are both in its connectivity set and neighborhood set. The higher the score, the better the neighborhood set of s_i reflecting the true structure of its connectivity set. If we treat the data subjects in the neighborhood set of each individual to be the friend it recognizes, then for a data subject with large popularity score, it means that most of people it recognizes as "friends" also recognize it as their friends. Thus, this subject is "popular".

For a fixed k, a *local asymmetric network (LAN)* of degrees k is constructed with the k-neighborhood set and k-connectivity set of each individual. The outward and inward edges of subject s_i is defined as

$$\mathcal{O}(s_i) = \{d_{ij} \mid s_j \in \mathcal{N}_k(s_i)\}, \quad \mathcal{J}(s_i) = \{d_{ij} \mid s_j \in C_k(s_i)\}. \tag{4.2}$$

Figure 4.2 shows two examples of such a network (with $k = 3$). Note that in Figure 4.2a, subject s_1 and s_2 have one inward edge each, while other subjects have more. In addition, the outward edges of s_1 and s_2 are much longer than the inward edges. For Figure 4.2b, subject s_1 to s_4 have more inward edges than the rest, and the inward edges are also much larger than the outward edges. From the examples it is seen that such a LAN can be used to detect the outliers. If we treat the directed edges in LAN as recognized "friends" (as in social network), then the outliers in the left figure recognize friends who do not recognize

them as friends, and the small clusters in right figure are "popular" people who do not recognize the people who recognize them as friends. Such a construction is different from previous density based approach like LOF. While s_1 and s_2 in the left figure have small data depth, the small clusters in the right figure are in the center of the data set and are much more difficult to detect.



(a) Example one of LAN. (b) Example two of LAN.

Figure 4.2: LAN examples.

The set of inward and outward edges have the following property:

Proposition 4.2.1. *For subject $s_i \in \mathcal{D}$ $(i = 1, \ldots, n)$, its inward edges set is composed of two non-overlapped sets, i.e. $\mathcal{J}(s_i) = \tilde{O}(s_i) \bigcup \tilde{\mathcal{J}}(s_i)$ where $\tilde{O}(s_i) \subset O(s_i)$, $\tilde{\mathcal{J}}(s_i)$ satisfies $\forall d \in \tilde{\mathcal{J}}(s_i)$, $d > d^*$, where $d^* = \max_{d_{ij}} O(s_i)$.*

This indicates that any subject's inward edges are either equal to some edges in its neighborhood set, or larger than any edges in its neighborhood set.

4.2.2 Preliminary

Before we introducing the formal definition of LoCO score, we first present some intuitive understanding about the idea. For each subject in $s_i \in \mathcal{D}$ $(i = 1, \ldots, n)$ and a specified neighborhood value k, we can calculate the empirical cumulative distribution functions of its inward and outward edges. Denote them as $F_{in}^{(k)}(d)$ and $F_{out}^{(k)}(d)$ respectively. It is

naturally to quantify the outlyingness of $\boldsymbol{s}_i$ by comparing the difference between these two empirical distributions. A classical test to do this is Kolmogorov-Smirnov (KS) test (Massey Jr (1951), Dimitrova, Kaishev, and Tan (2017)). It is a nonparametric test of the equality of continuous or discontinuous one-dimensional probability distributions. The test statistic is

$$T_k(\boldsymbol{s}_i) = \sup_d |F_{in}^{(k)}(d) - F_{out}^{(k)}(d)|, \tag{4.3}$$

which is the maximum of these two empirical difference. Intuitively, if F_{in} and F_{out} come from the same distributions for each sample, then the corresponding score would be close to 0. Similarly, if the two distributions are quite different, then the resulting score would be close to 1. This would be a natural way to quantify the degree of outlyingness at the first glimpse. The resulting score actually has an upper bound. We summarize it in the following theorem:

Theorem 4.2.2. *Let $\boldsymbol{s}_i$ be any subject from the data $\mathcal{D}$. Denote $|\tilde{O}(\boldsymbol{s}_i)| = \tilde{m}_k^{(i)}$, $|\tilde{\mathcal{J}}(\boldsymbol{s}_i)| = \tilde{q}_k^{(i)}$. Then,*
(i). If $m_k^{(i)} \geq q_k^{(i)}$, $T_k(\boldsymbol{s}_i) \leq 1 - \frac{\tilde{m}_k^{(i)}}{m_k^{(i)}}$;
(ii). If $m_k^{(i)} < q_k^{(i)}$, $T_k(\boldsymbol{s}_i) \leq 1 - \frac{\tilde{m}_k^{(i)}}{q_k^{(i)}}$.

The proof of the theorem can be found in Appendix A.5. It can be seen that the upper bound is constrained by the total number of elements that are both contained in $O(\boldsymbol{s}_i)$ and $\mathcal{J}(\boldsymbol{s}_i)$. As a result, given a data subject $\boldsymbol{s}_i$, if there are lots of common subjects in its connectivity and neighborhood sets, the corresponding $T_k(\boldsymbol{s}_i)$ will be bounded by a smaller value. Thus, it would be less possible for this point to be an outlier.

Although the whole idea seems straightforward, when we run simulation examples based on this idea, it will give us some counter-intuitive results in some scenarios. The main reason behind this is because it is not easy to quantify the inward and outward edges differences across different data subjects consistently. Thus, base on this idea, we came

up with the local connectivity score which can properly quantify the difference. We will introduce the LoCO score in the next section.

4.2.3 LoCO Score

Assuming for a normal data subject, the length of its inward and outward edges generated from the connectivity and neighborhood set follows the same distribution. Based on the definitions and notations we introduced above, we define the *LoCO score* for subject $\boldsymbol{s}_i$ as:

$$O_k(\boldsymbol{s}_i) = \begin{cases} \frac{\sum_{s_j \in \mathcal{N}_k(s_i) \setminus C_k(s_i)} Pop(\boldsymbol{s}_j)}{\sum_{s_j \in \mathcal{N}_k(s_i) \cup C_k(s_i)} Pop(\boldsymbol{s}_j)} & \text{if } C_k(\boldsymbol{s}_i) \neq \emptyset, \\ 1 + den(\boldsymbol{s}_i) \quad C_k(\boldsymbol{s}_i) = \emptyset. \end{cases} \tag{4.4}$$

Here, $den(\boldsymbol{s}_i)$ is defined as

$$den(\boldsymbol{s}_i) = \frac{1}{\left|\mathcal{N}_k(\boldsymbol{s}_i)\right|} \sum_{\boldsymbol{s}_j \in \mathcal{N}_k(\boldsymbol{s}_i)} d_{ij}/C \tag{4.5}$$

where C is a pre-chosen large constant. We add the $den(\boldsymbol{s}_i)$ to differentiate subjects with empty connectivity set. As a result, the outlying score will range in $[0, 1]$ when $C_k(\boldsymbol{s}_i) \neq \emptyset$, and otherwise larger than 1.

LoCO score can be interpreted as the weighted proportion of subjects which only belong to $\boldsymbol{s}_i$'s neighborhood set, but not belong to its connectivity set. Furthermore, if those points which only belong to the neighborhood set are popular, it makes the overall score larger. It confirms with our intuition that if popular neighborhood points of a subject are not contained in the corresponding connectivity set, then the respective subject should be more likely to be isolated, thus having a higher outlying score. If we treat $\boldsymbol{s}_i$'s neighborhood as its friend, it indicates although $\boldsymbol{s}_i$ treat some "popular" person (subject with larger scores) as its friends, they don't recognized $\boldsymbol{s}_i$ as "friend" sadly. Overall, this quantity tells us how similar the subject's neighborhood set and connectivity set are. As we can imagine, the larger the value, the more isolated $\boldsymbol{s}_i$ becomes.

On the other hand, we can interpret this score as a similarity quantity between two empirical distributions. Under the assumption that a normal data subject has similar neighborhood and connectivity set, for any subject $\boldsymbol{s}_j$, we can estimate the empirical distribution of its inward edges' length as

$$P(x = d) = \frac{Pop(\boldsymbol{s}_{j'})}{\sum_{\boldsymbol{s}_j \in \mathcal{N}_k(\boldsymbol{s}_i) \bigcup C_k(\boldsymbol{s}_i)} Pop(\boldsymbol{s}_j)}, \quad \text{s.t. } d_{ij'} = d, \tag{4.6}$$

where $d \in \mathcal{O}(\boldsymbol{s}_i) \bigcap \mathcal{J}(\boldsymbol{s}_i)$. But the truth is that only values in $\mathcal{J}(\boldsymbol{s}_i)$ are generated from the true distribution of inward edge length. Thus, we get another empirical distribution by suppressing all distances not belong to the inward edges to zero. Namely,

$$P^*(x = d) = \begin{cases} P(x = d), & \text{if } d \in \mathcal{J}(\boldsymbol{s}_i), \\ 0, & \text{otherwise} \end{cases} \tag{4.7}$$

where $d \in \mathcal{O}(\boldsymbol{s}_i) \bigcap \mathcal{J}(\boldsymbol{s}_i)$. Obviously, the values above cannot sum to 1. Since the observed connectivity set should be representative to the corresponding length distribution, we can assume other length values with non-zero probability should be larger than those include in $\mathcal{J}(\boldsymbol{s}_i)$, and all of these lengths' probability sum to 1.

Once we get two version of estimated empirical distribution of the inward edge length, denote as $P(x)$ and $P^*(x)$, the LoCO score can actually be interpreted as the KolmogorovSmirnov test statistic (Dimitrova, Kaishev, and Tan 2017) between these two empirical distribubtions.

Due to the fact that LoCO score accounts for the directed relationship between the inward and outward edges for each individual, the resulting outlying score will automatically reflect the effect of density changes. Back to the example shown in Figure 4.1, when $k = 3$, subject A's connectivity outlying score is 0.429, which is much lower compared with subject B with connectivity outlying score 2.003. As a result, LoCO score can correctly reflect the density changes within the data based on the neighborhood points it considers (k). We

will illustrate its stableness in terms of performance through numerical examples later.

4.2.4 Maximum Outlying Score

A lot of local outlier detection methods like LOF will suffer from sensitivity of choosing neighborhood values k. For example, for local outlier factor, when k changes, the quotient values between the local density of the target point and its neighbors will both vary. So it is almost impossible to give a consistent threshold to claim one point to be an outlier. On the contrary, for connectivity outlying score, when the outlying score is large, the corresponding subject would always be an outlier consistently regardless of any specific scenarios or ks. On the other hand, it is easy to see that the LoCO score for one subject will have a decreasing trend as k increases overall. As a result, it will not be necessary to choose very large k in terms of detecting "local" outliers. Given those facts, we can define a maximum outlying score for each subject within a pre-specified neighborhood range.

Definition (Maximum outlying score within $[k_1, k_2]$)
Given a dataset $\mathcal{D}$ with n subjects $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$, for any $k \in [k_1, k_2]$, we can calculate the corresponding LoCO score for each k in this range. Then the maximum outlying score within $[k_1, k_2]$ is defined as

$$O^*(\boldsymbol{s}_i) = \max_{k \in [k_1, k_2]} O_k(\boldsymbol{s}_i), \tag{4.8}$$

where $O_k(\boldsymbol{s}_i)$ is LoCO score defined in equation (4.4).

We choose the neighborhood values ks within a pre-specified range mainly for two reasons. First of all, looping through all possible ks would be computationally expensive and unnecessary. Because when k is too large, the difference between each subject's directed networks is weakened. The resulting outlying score will decrease in general. On the other hand, if k is too small, then lots of points will seem to be an outlier within its neighborhood. As a result, we need to choose k neither too big, nor too small. For example, for a dataset

with 200 samples, choosing k from 5 to 20 might be enough.

Maximum outlying score is derived from LoCO score based on its based on its consistent outlyingness property so that we can clearly judge whether the score is large or not without worrying about varying settings. It is motivated from the purpose of not specifying a unique neighborhood values. In the next section, we introduce "p-value" induced by the LoCO score. It will have clearer statistical interpretation, and it can also be generalized using the same idea in this section to reduce the demand to specify the neighborhood value k.

4.3 Outlier Detection with Confidence: a "p-value" for LoCO Score

The LoCO score can be incorporated into the framework of conformal prediction. Conformal prediction is a prediction framework with theoretically guaranteed confidence level. Is is developed based on a state-of-art non-parametric predictive inference tool in machine learning and statistics, known as *conformal prediction* (Vovk, Gammerman, and Shafer 2005, Lei et al. 2018, Barber et al. 2019b, Barber et al. 2019a, Vovk, Gammerman, and Shafer 2005, Xie and Zheng 2020a). Specifically, given a data set with IID samples $\mathcal{D} = \{\boldsymbol{s}_i \in \mathbb{R}^p \mid i = 1, \ldots, n\}$, a new sample $\boldsymbol{s}_{n+1}$ is considered to be an outlier if it is not drawn independently from the same distribution. Our goal is to judge whether this new sample is an outlier or not.

Denote the dataset including the newly added sample as $\mathcal{D}^* = \mathcal{D} \bigcup \{\boldsymbol{s}_{n+1}\}$. Assume $\boldsymbol{s}_{n+1}$ is drawn independently from the same distribution within the data. Namely, $\boldsymbol{s}_i \sim \mathcal{F}$ $(i = 1, \ldots, n+1)$. Define *conformal matrix* constructed by the conformal score of $\boldsymbol{s}_i$ w.r.t. $\boldsymbol{s}_j$ as $R_{i,j} = O^{(-j)}(\boldsymbol{s}_i)$ $(1 \leq i \neq j \leq n+1)$ where $O^{(-j)}(\boldsymbol{s}_i)$ is the outlying score of $\boldsymbol{s}_i$ for some k within $\mathcal{D}/\{\boldsymbol{s}_i, \boldsymbol{s}_j\}$.

Note that for $R_{n+1,i} = D^{(-i)}(\boldsymbol{s}_{n+1})$, it is defined as the outlying score of $\boldsymbol{s}_{n+1}$ within $\mathcal{D}_{i,n+1} = \mathcal{D}^*/\{\boldsymbol{s}_{n+1}, \boldsymbol{s}_i\}$. Thus, it is actually a map from $s_{n+1} \oplus \mathcal{D}_{i,n+1}$ to $\mathbb{R}$. Similarly, $R_{i,n+1} = O^{(-(n+1))}(s_i)$ constructs a map from $\boldsymbol{s}_i \oplus \mathcal{D}_{i,n+1}$ to $\mathbb{R}$. Under the assumption that

$\boldsymbol{s}_{n+1}$ is drawn independently from the same distribution within the data, there exists distributional symmetry between $\boldsymbol{s}_i$ and $\boldsymbol{s}_{n+1}$. It naturally induces the distributional symmetry of $R_{n+1,i}$ and $R_{i,n+1}$. Namely, we have $R_{i,n+1} \sim R_{n+1,i}$.

For each s_{n+1}, we define the conformal score

$$V(\boldsymbol{s}_{n+1}) = \frac{1}{n}\sum_{i=1}^{n} 1\{R_{n+1,i} \leq R_{i,n+1}\} \tag{4.9}$$

which relates to the degree of "conformity" of score values $R_{n+1,i}$ among scores of $R_{i,n+1}$. Specifically, if $V(\boldsymbol{s}_{n+1}) \approx \frac{1}{2}$, then $R_{n+1,i}$ is around the middle of the original pool of LoCO score $R_{i,n+1}(i = 1, \ldots, n)$. When $V(\boldsymbol{s}_{n+1}) \approx 0$, $R_{n+1,i}$ is at the extreme end of the LoCO score $R_{i,n+1}$ and thus "least conformal". This leads the following definition of the conformal region

$$C_\alpha = \{x : V(x) \geq \alpha\}. \tag{4.10}$$

The following theorem states that, under the IID assumption, C_α defined in (4.10) is guaranteed a level $1 - \alpha$ for a "ordinary" $\boldsymbol{s}_{n+1}$. This can be viewed as a variant version of *Jackknife-plus predictive interval* proposed by Barber et al. 2019a.

Theorem 4.3.1. *Suppose* $\{\boldsymbol{s}_i\}_{i=1}^n$, $\boldsymbol{s}_{n+1} \overset{i.i.d.}{\sim} \mathcal{F}$. *Then we have* $P(\boldsymbol{s}_{n+1} \in C_\alpha) \geq 1 - \alpha$.

Theorem 4.3.1 is proved in Appendix A.7 for a finite n with a conservatively guaranteed converge rate of $1 - \alpha$. We treat C_α as an approximate level-$(1 - \alpha)$ confidence interval.

Inspired by equation (4.9), we can treat $V(\boldsymbol{s}_{n+1})$ as the p-value against the null hypothesis: $\boldsymbol{s}_{n+1}$ is not a local outlier. The smaller the p-value, the stronger the evidence of rejecting the null hypothesis.

Motivated by the maximum outlying score defined in Section 4.2.4, we can also define minimum conformal score based on the "p-value" defined in equation (4.9). Calculating the conformal score might be slower than LoCO score. As a result, we may choose which

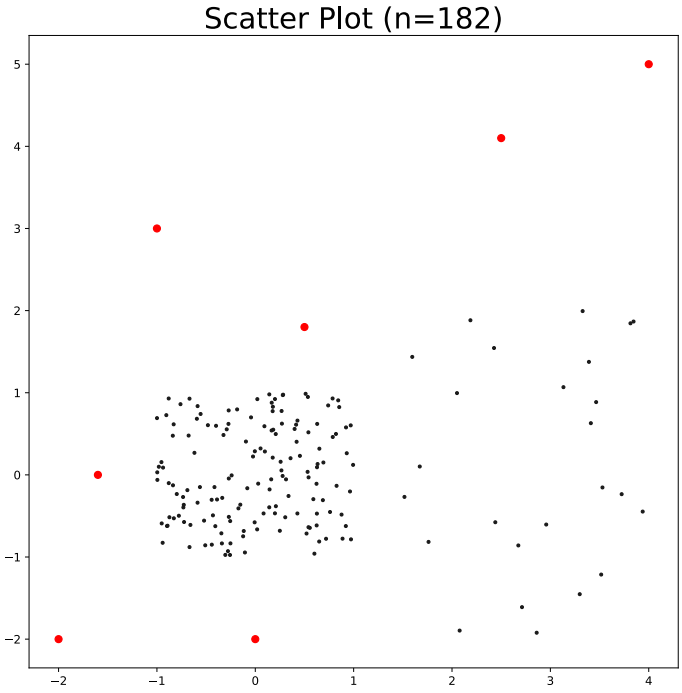


Figure 4.3: Example one (scenario one): scatterplot.
There exist two clusters in the plot, and the densities of the two clusters are quite unbalanced.

to use based on our specific needs in real practice.

Up to now, we have introduced the LoCO score based on LAN formulated by each subject's neighborhood. Then we derived two variants of the LoCO score: maximum outlying score and conformal outlying score to improve it from different aspects. In the next section, we will conduct series of experiments to illustrate the performance of the proposed methods.

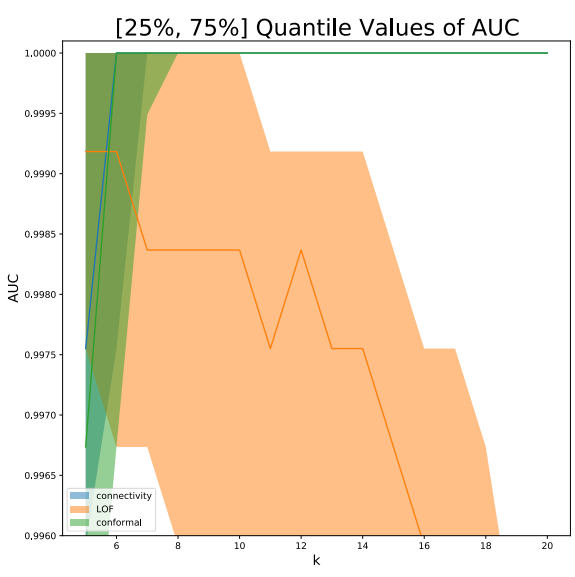
4.4 Simulation Examples

Example one: scenario one

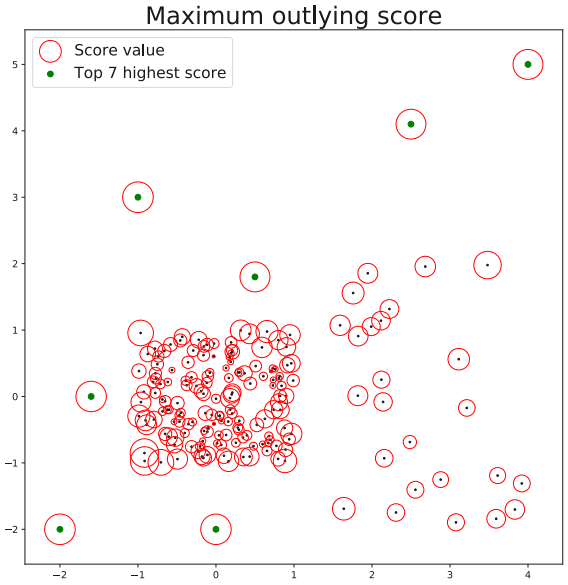
In this example, we explore a dataset with 182 points. Figure 4.3 shows the scatter plot of the data with 7 outliers marked in red. There are two clusters of points in the data marked in black with different densities generated from uniform distribution. We repeat the experiments for 100 times and calculate the corresponding outlying scores respectively.

We first compare the performance of the three methods with different neighborhood values: LoCO score, conformal outlying score and LOF. Figure 4.4a shows the AUC of different methods with neighborhood values k from 5 to 20. It can be seen that the LoCO score in blue and conformal outlying score in green have very similar performance, and their AUC values increase to 1 as k increases to 8. It means that they will perfect detect all outliers as k increases. For LOF, the $[25\%, 75\%]$ quantile range is very large, and has a decreasing trend within this range. Thus, our method has better and more stable performance in this scenario.

Furthermore, we calculate the maximum outlying score within the range $[8, 20]$ for all of the 100 experiments. The resulting AUC among the 100 experiments has median value 1 and standard deviation 0.00046. This indicates that the maximum outlying score perfectly identified the outlier for most of the times. Figure 4.4b shows the resulting scores corresponding to the median AUC values, and the radius of the red circle stands for the score. The outlier in green corresponds to the top 7 biggest circles perfectly.



(a) AUC of different methods.



(b) Maximum outlying score within $[8, 20]$.

Figure 4.4: Example one (scenario one): results.

On the other hand, we can calculate the minimum conformal score based on the confor-

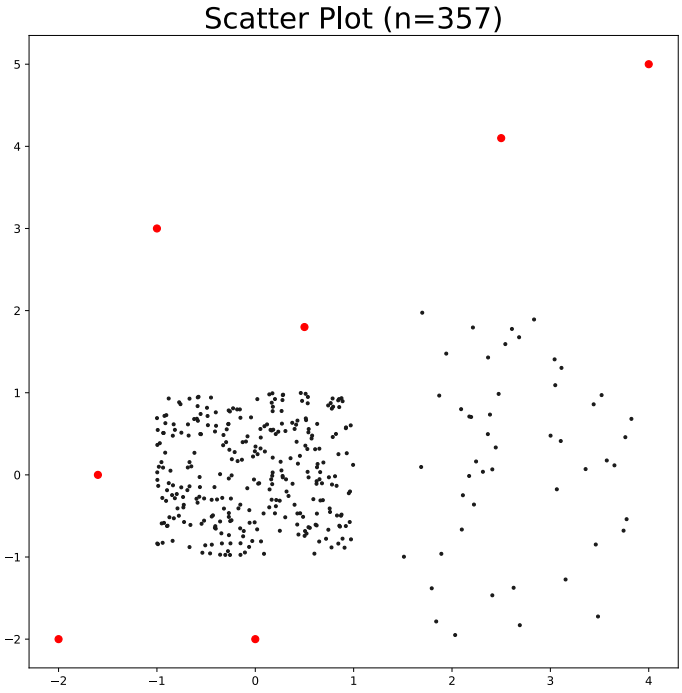


Figure 4.5: Example one (scenario two): scatterplot.
We doubled the sample size for the two clusters compared with scenario one.

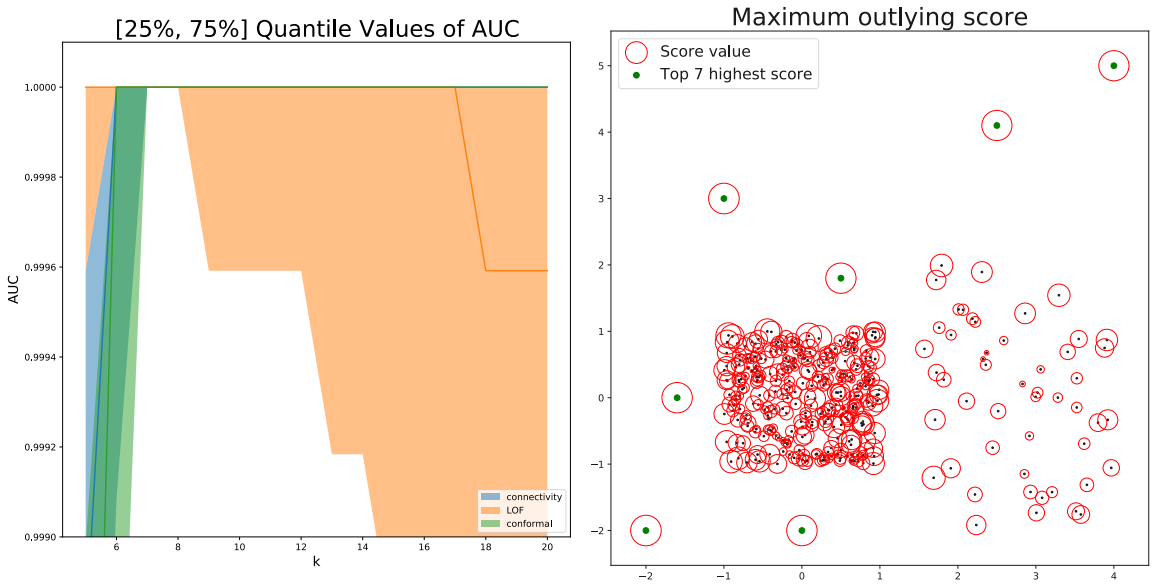
mal outlying score within the range $[8, 20]$. The corresponding median AUC and standard deviation are 1 and 0.0010. The results are quite similar to maximum outlying score.

Example one: scenario two

In the second scenario, we double the sample size of the two clusters. The scatter plot of the data is shown in Figure 4.5. We also repeat the experiments for 100 times.

Figure 4.6a shows the results of the three methods. It can be seen that same as before, the AUC values from LoCO scores and conformal outlying scores are quite similar, and AUC for local outlier factor is not vary stable compared with the other two methods. But since the sample size was doubled from the first scenario, the density of the two clusters are larger. Thus, LOF also has a small range of k having AUC values to be exactly 1. But overall, our proposed methods are still more stable in this scenario.

Figure 4.6b shows the resulting maximum outlying score. The median and standard deviation of the AUC values among the total 100 experiments is 1 and 0.00038. Thus, for most of the experiments, maximum outlying score will also identify the outliers perfectly.



(a) AUC of different methods.

(b) Maximum outlying score within $[8, 20]$.

Figure 4.6: Example one (scenario two): results.

We also calculate the minimum conformal score. The median AUC is also 1 with standard deviation 0.0006, slightly larger than maximum outlying score.

Example one: scenario three

In the third scenario, we make the density within the two clusters almost equal. The scatter plot of the data is shown in Figure 4.7. We also repeat the experiments for 100 times.

Figure 4.8a shows the results of the three methods. It can be seen that all of the three methods perform very well in this scenario. The main reason is because there is no serious cluster density change within the data.

Figure 4.8b shows the resulting maximum outlying score. The median and standard deviation of the AUC values among the total 100 experiments is 1 and 0.00018. We also calculate the minimum conformal score. Same as before, the median AUC value is 1, and the standard deviation is 0.0004 which is also slightly larger than maximum outlying score.

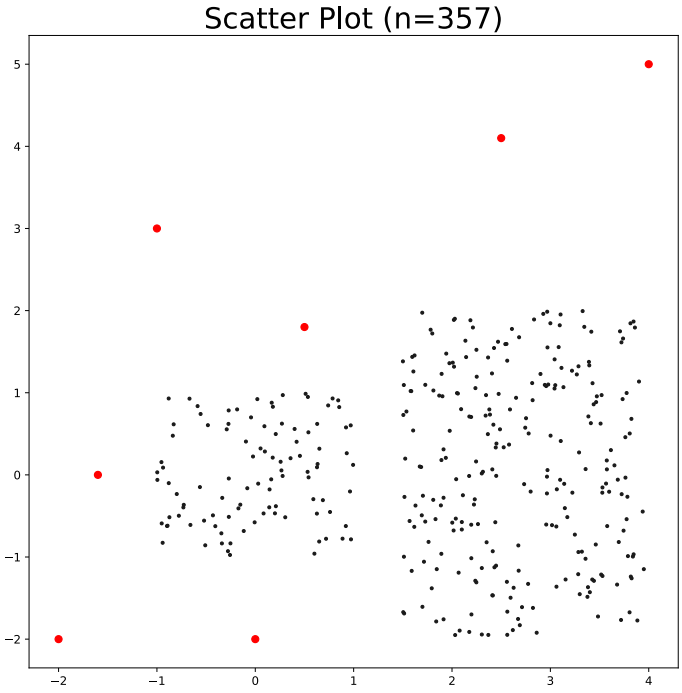


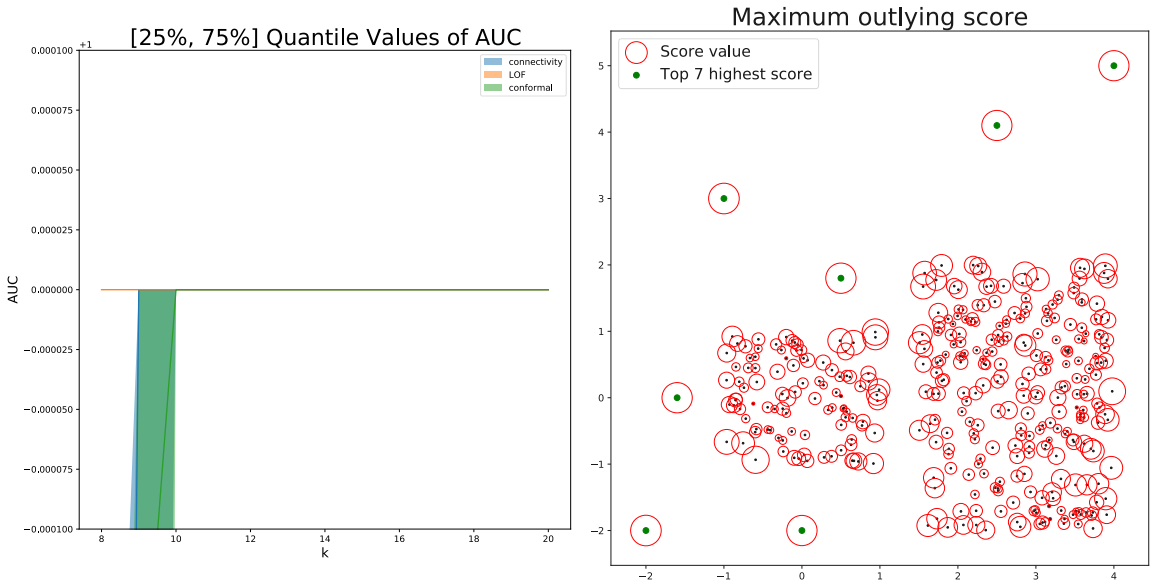
Figure 4.7: Example one (scenario three): scatterplot.
We make the two clusters to have almost equal densities in this scenario.

4.4.1 Example Two: Outliers in the Center of a Circle

In this example, we consider a data with outliers within a circle. The total sample size is 504. Since these outliers are close to the centroid of the data, it would be difficult for depth based outlier detection techniques to detect them. The scatter plot of the data is shown in Figure 4.9. We also repeat the experiments for 100 times.

Figure 4.10a shows the results of the three methods. It can be seen that the AUC values from LoCO scores and conformal outlying scores are still quite similar, and AUC for local outlier factor will decrease quickly as k increases.

Figure 4.10b shows the resulting maximum outlying score. The median and standard deviation of the AUC values among the total 100 experiments is 1 and 0.00078. Thus, for most of the experiments, maximum outlying score will also identify the outliers perfectly. We also calculate the minimum conformal score. The median AUC value is 1 with standard deviation 0.0010.



(a) AUC of different methods.

(b) Maximum outlying score within $[8, 20]$.

Figure 4.8: Example one (scenario three): results.

4.5 Real Data Examples

4.5.1 AIS Vessel Data

The Automatic Identification System (AIS) is an automated, autonomous tracking system which is extensively used in the maritime world for the exchange of navigational information between AIS-equipped terminals. Vessel Traffic Services (VTS) ashore use AIS to identify, locate and monitor vessels. The Panama Canal uses the AIS as well to provide information about rain along the canal as well as wind in the locks. Thus, it provides us with large amount of data for analysis. We will use the proposed method to detect abnormal maritime traffic events using the AIS data.

The features under study are:

- Maritime Mobile Service Identity (MMSI): a unique identification number;

- Longitude and Latitude: vessel's location at the recorded time stamp;

- Vessel type: two/four digits code indicating vessels type. For example: 1004 for cargo, 1001 for fishing etc.

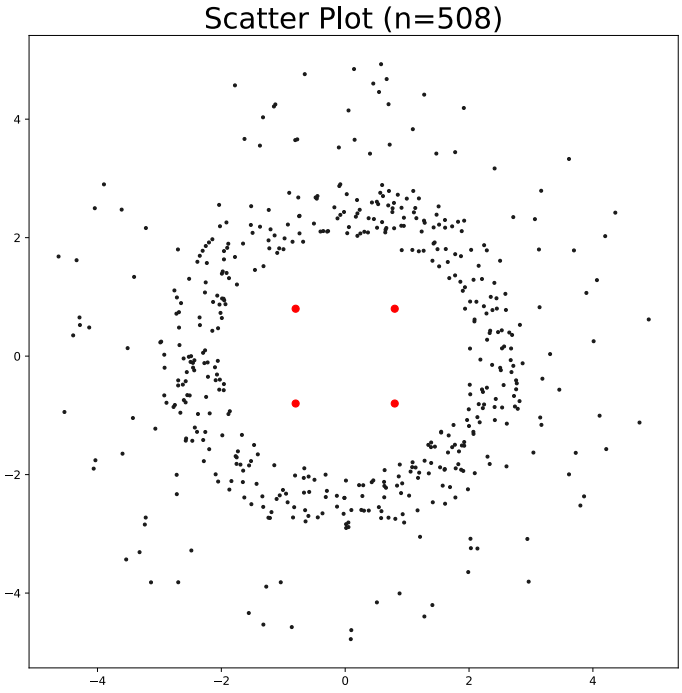
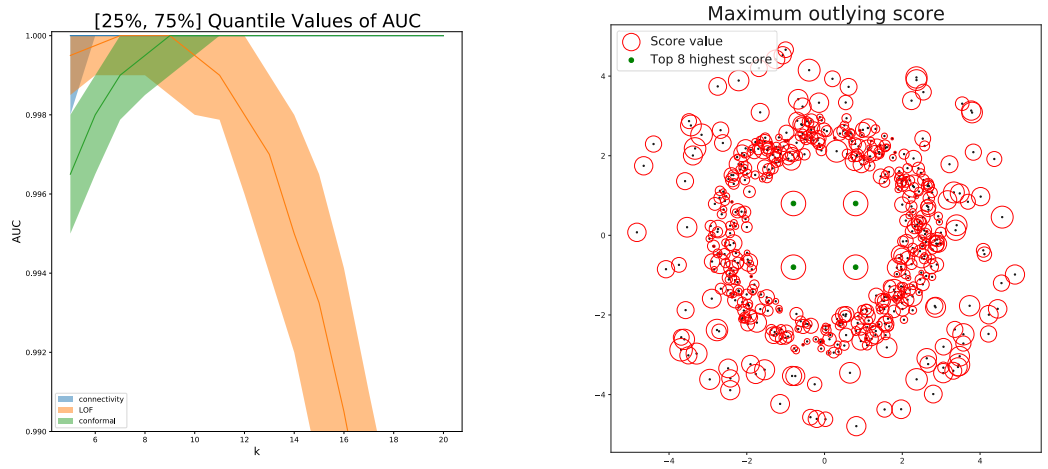


Figure 4.9: Example two: scatterplot.
We draw a ring with sparse density around the outer margin. Four outliers are located in the center.

The data is available at `https://marinecadastre.gov/ais/`. Due to the fact that each vessel will send signals about every one or two minutes while in route. The data we are dealing with is massive. For example, US coast and waterway related information accounts for 32 GB of AIS data each day. As a result, the trajectory of different vessels within a certain area could be messy if the duration is long. For example, Figure 4.11a plots cargos' trajectory within a small box around Newark port in Zone 18 between June 2017 to December 2017 (Six months). We can see that the vessels' trajectories spread over the whole box.

Since lots of suspicious and illegal behaviors could happen around vessels entering the US maritime limits boundaries and heading to some sea port, we will consider inbound vessels heading to Newark within the half year between June 2017 to December 2017 and restrict the trajectory within a geometric box containing Newark. We only consider the vessel trajectory (longitude and latitude) information for now. Figure 4.11b shows the inbound trajectories within this period. There are 630 valid vessels in total within this



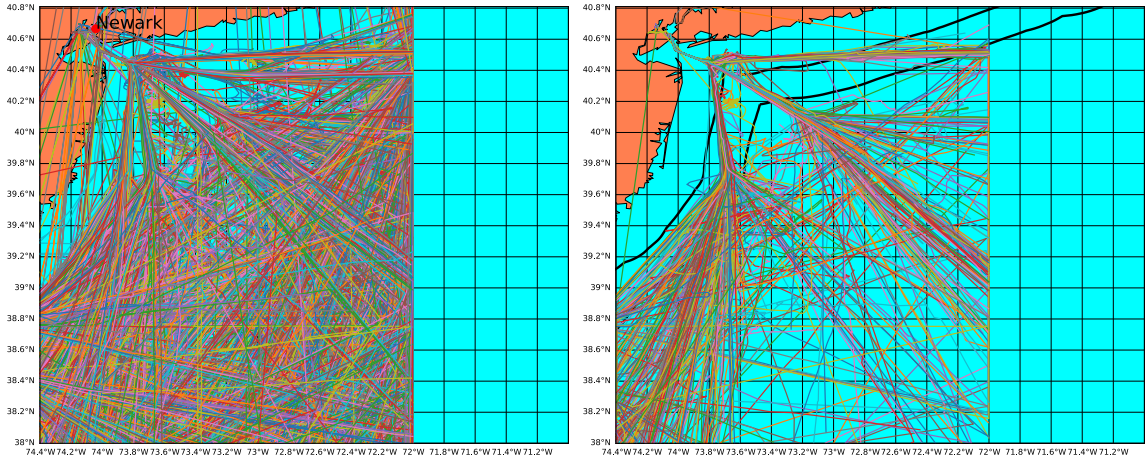
(a) AUC of different methods.

(b) Maximum outlying score within $[8, 20]$.

Figure 4.10: Example two: results.

period.

We use dynamic time warping (DTW) distance (Sakoe and Chiba 1978) to measure the similarity between different trajectories. Since we do not want to specify a unique k here, we just use the maximum outlying score introduced in Section 4.2.4 to measure the outlyingness of each sample. We calculate the maximum outlying score from $k = 15$ to $k = 55$ at the interval of 5 ($k = 15, 20, \ldots, 55$). There are 42 vessels with maximum outlying score larger than 1 in total. We plotted the vessel trajectories with top 21 LoCO scores in Figure 4.12. The red line stands for each vessel's trajectory, and the colored dashed lines around the red line stand for the top 60 nearest neighborhood vessels trajectories of the current vessel. It can be seen that all of the vessels seem to behave differently compared with their neighborhood vessels trajectories. Some of the abnormal behavior only happens within a small part in the trajectory. For example, Vessel 266, 228 and 169 seem to have some twisted behavior only in a local region. On the contrary, some other vessels will deviate from the normal pattern within its neighbors severely. For example, Vessel 162 or Vessel 121. This indicates that our method can detect different types of abnormal behaviors for the vessel data.



(a) Vessel trajectory. (b) Inbound vessel trajectories to Newark.

Figure 4.11: Vessel trajectories.

4.5.2 Email Network Data

We analyze an email data from a large European research institution. We have anonymized information about all incoming and outgoing email between members of the research institution. The e-mails only represent communication between institution members (the core), and the dataset does not contain incoming messages from or outgoing messages to the rest of the world. We analyze the "data email-Eu-core-temporal-Dep1" data containing 309 nodes (users) along 75 consecutive weeks (more than 1 year). The data can be obtained from `http://snap.stanford.edu/data/email-Eu-core-temporal.html`. It contains the following features

- SRC: id of the source node (a user), ranging from 1 to 309;

- TGT: id of the target node (a user), ranging from 1 to 309;

- TS: timestamp (in seconds), starting from 0.

We will analyze the weekly behavior of the 309 nodes using our proposed method. Since the accurate date information of this data is unknown. We manually cut the data into 75 pieces with each piece containing 7 consecutive days events. Each piece is treated as a "week". For each week, we construct a counting matrix $\boldsymbol{A}_t = \{a_{ij}\} \in \mathbb{R}^{309\times309}$ ($t = 1\ldots,75$), a_{ij} stands for the total number of counts of sending and receiving emails between

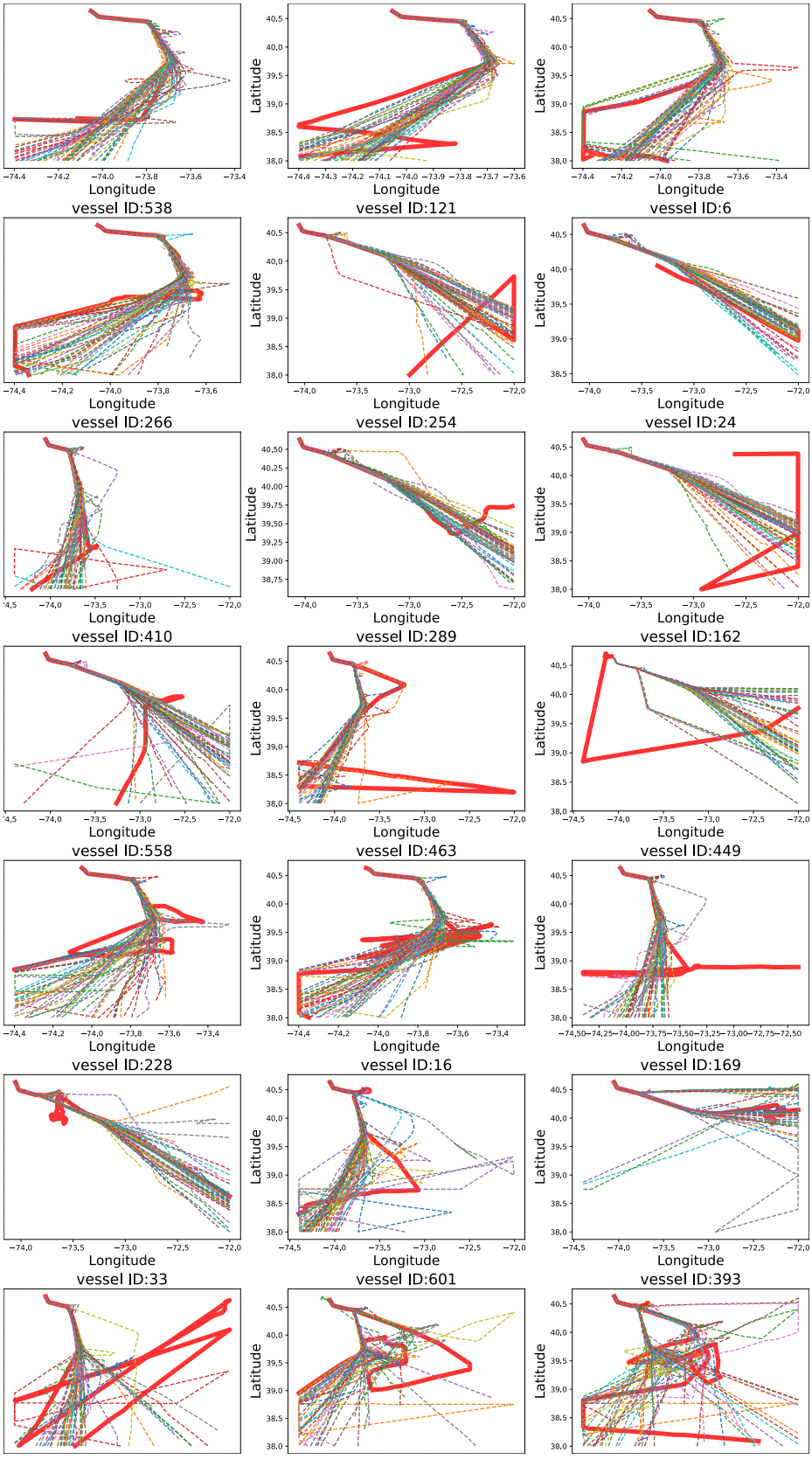


Figure 4.12: Top 21 abnormal vessels.

user i and user j ($1 \leq i \neq j \leq 309$). We will explore the weekly counting matrices and find out the abnormal weeks using the proposed methods.

In order to calculate the outlying score, we need similarity measure between the counting matrix of each week. For any symmetric matrix $\boldsymbol{A}_1 = \{a_{ij}^{(1)}\}$, $\boldsymbol{A}_2 = \{a_{ij}^{(2)}\} \in \mathbb{R}^{n\times n}$ with $n \in \mathbb{N}^+$, We introduce 3 distance measure of matrices.

Definition (correlation matrix distance) Herdin et al. 2005 introduced a correlation matrix distance as below

$$d = 1 - \frac{tr(\boldsymbol{A}_1 \cdot \boldsymbol{A}_2)}{\|\boldsymbol{A}_1\|_F \cdot \|\boldsymbol{A}_2\|_F},$$

where $\|\cdot\|_F$ denote the Frobenius norm. It is equivalent to one minus cosine similarity between these two matrices.

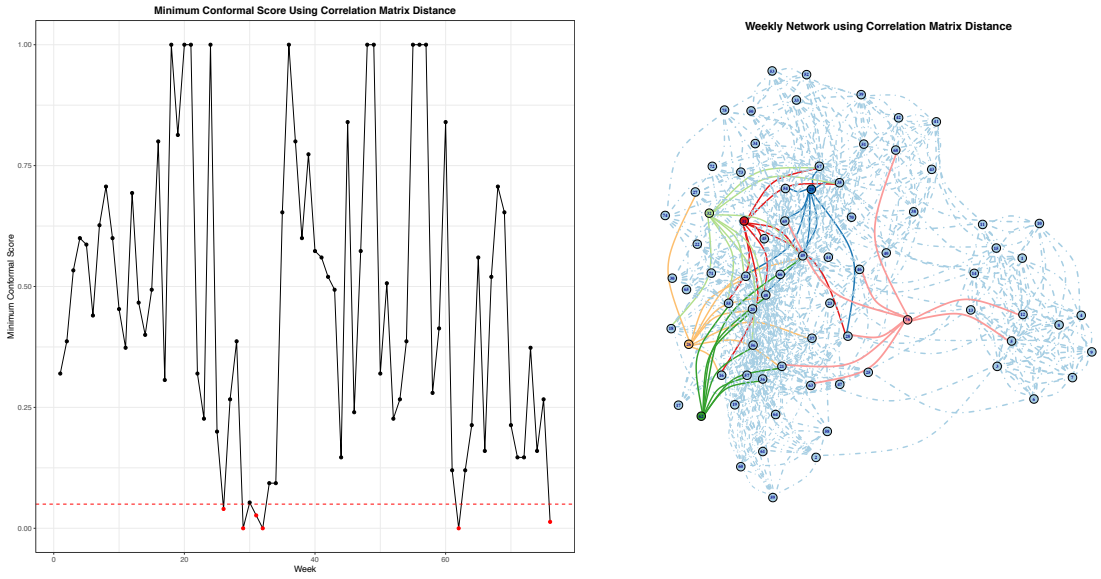
Definition (Frobenius norm distance) The Frobenius norm distance between $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ is defined as

$$d = \|\boldsymbol{A}_1 - \boldsymbol{A}_2\|_F.$$

Definition (L_0 norm distance) Generalized from the L_0 vector norm, the L_0 norm distance between $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ is defined as the total number of non-zeros cells in $\boldsymbol{A}_1 - \boldsymbol{A}_2$.

We calculated the minimum conformal scores using the three distance measures within the range $[8, 20]$. For correlation matrix distance. There are 6 weeks with minimum conformal score smaller than 0.05. Figure 4.13a shows the corresponding scores for each week. It can be seen that the abnormal weeks concentrate on weeks within $25 - 30$ and $60 - 70$. Figure 4.13b shows the network constructed by the corresponding distances. the network has some clustered patterns, and some early weeks forms a cluster on the lower right of the plot. The node in color stands for the detected abnormal points. Each node connects with its 8-nearest neighborhood nodes. It can be seen that all these outliers are very isolated with empty connectivity sets.

For the Frobenius norm distance, we also calculate the conformal outlying score within $[8, 20]$. There are 5 weeks will scores smaller than 0.05. Figure 4.14a shows the cor-



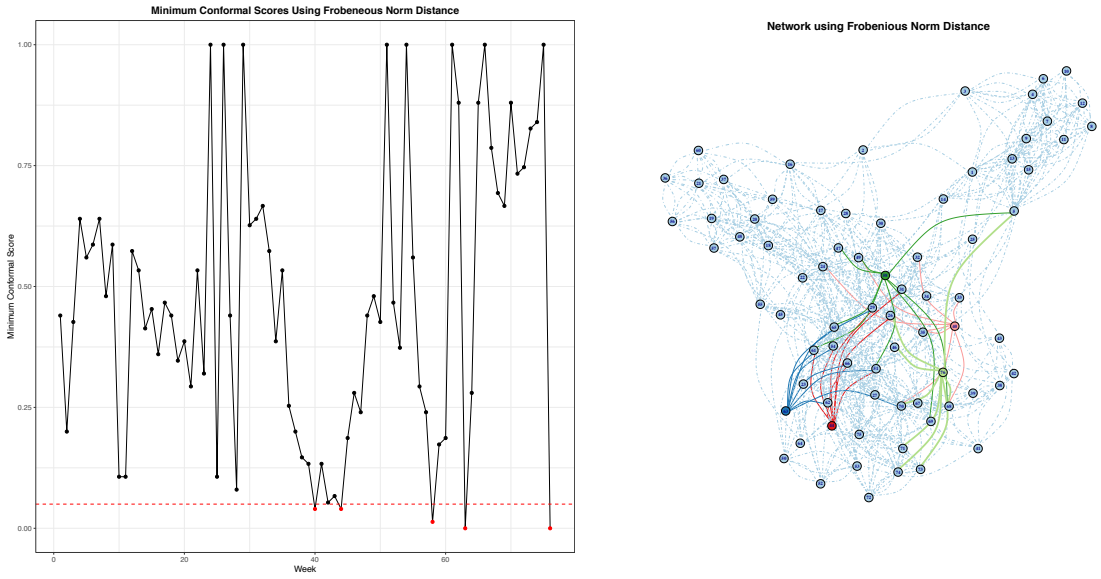
(a) Maximum outlying score for each week. (b) Network for correlation matrix distance

Figure 4.13: Correlation matrix distance: results.

responding scores for each week. It can be seen that the abnormal weeks differs a little from the previous one, and it mainly concentrates on the mid-period and late-period of the whole range. Figure 4.14b shows the network constructed by the corresponding weeks. It has some different clustered pattern, with early, middle and late phase forming three different clusters. The detected outliers are very isolated with empty connectivity sets.

For the L_0-norm distance, there are 11 weeks with minimum conformal score smaller than 0.05. All of these weeks stay at the late-phase of the whole period. Figure 4.15b shows the network constructed by the corresponding weeks. It can be seen that the shape of this network differs a lot from the previous two measurements. It has a clear center, and all the rest of the subjects choose the subjects in the center as their neighborhood. Thus, it forms a "flower" shape cluster. It indicates that there exist some weeks (corresponding to the subjects in the center) which have very similar pattern to the rest of the weeks.

In summary, from the email data, we can see the capability of using the connectivity outlying score to detect abnormal behaviors within network data. Since the information of the data is limited and we do not have the exact date information, we cannot directly justify



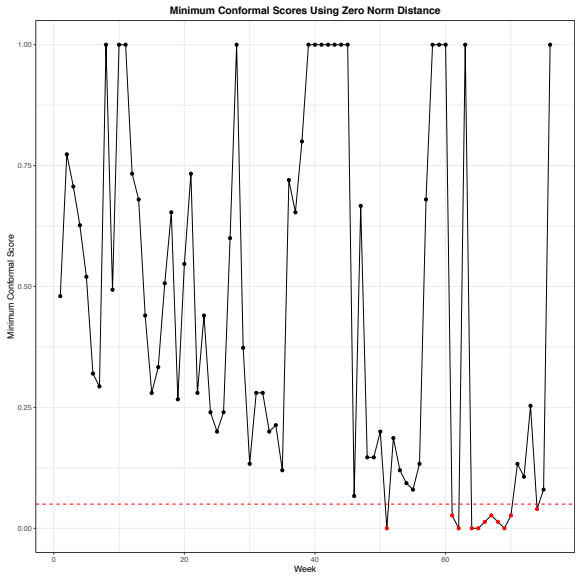
(a) Maximum outlying score for each week. (b) Network for Frobenius norm distance.

Figure 4.14: Frobenius norm distance: results.

the reasonability of our detected subject in terms of the periodical effect. But we can still get some meaningful information from the network patterns we get from different distance metrics.

4.6 Conclusion

In this chapter, we have proposed a directed neighborhood-based local outlier detection method. The newly proposed local outlier detection method: LoCO method properly quantifies the degree of outlyingness of each subject in a data set through constructing the local asymmetric network (LAN). It is quite similar to LOF in terms of the output. But LOF tends to be hard to interpret in some cases. For example, an outlying score greater than 1 might not be an outlier in one data set, but it could have already been an outlier in another data set. Thus, the corresponding outlying scores are not quite consistent to different data set. In contrast to LOF, LoCO generate the outlying score based on the assigned popularity score of each subject, and it also consider the network effect within the neighborhood of each data subject. As a result, the corresponding score is more consistent and interpretable.



(a) Conformal outlying score at $k = 45$ for each week.

(b) Network for L_0-norm distance.

Figure 4.15: L_0-norm distance: results.

On the other hand, LOF will suffer when there exist clusters of points with varying densities. Since LoCO score relies on the network effect between each individuals, it will be more robust to the density changes as well. We perform a series of simulation examples to verify this statement.

We showed that LoCO has better performance in terms of AUC values in some scenarios which are difficult for LOF. The corresponding score are also easier to interpret. Furthermore, when we apply the proposed method into different simulation settings, it can be seen that the AUC values are quite robust to the changing of neighborhood values of k. We also developed a "p-value" based on the LoCO method, and it has better statistical interpretation.

We apply the proposed method onto two real data sets: the vessel data and email data. The detected abnormal trajectories are reasonable for the vessel data. For email data, we study the weekly email sending and receiving behavior through three different metrics. Different outliers are identified from them. There exist some seasonality patterns within the selected weeks.

CHAPTER 5

APPENDICES

A.1 Proof of Theorem 2.3.1

Proof. The proof will have some minor differences based on different distribution functions in GLM. We use Binomial distribution for logistic regression as an example. We begin with proving the first inequality. Equivalently, we need to prove

$$G(\boldsymbol{B}, \boldsymbol{B}^*) - f(\boldsymbol{B} \mid \boldsymbol{y}, \boldsymbol{X}) \geq \frac{\rho - \|\boldsymbol{H}\|_2/4}{2} \|\boldsymbol{B} - \boldsymbol{B}^*\|_F^2. \tag{5.1}$$

In order to prove the above equation, firstly we need to prove

$$\langle \sum \tau''(\langle \boldsymbol{X}_i, \boldsymbol{B} \rangle_F) \langle \boldsymbol{X}_i, \boldsymbol{\Phi} \rangle_F \boldsymbol{X}_i, \boldsymbol{\Phi} \rangle_F \leq \frac{1}{4} \|\boldsymbol{H}\|_2 \|\boldsymbol{\Phi}\|_F^2. \tag{5.2}$$

holds for any $\boldsymbol{\Phi} \in \mathbb{R}^{n \times p}$. For Binomial distribution, since $\tau(x) = \log(1 + e^x)$, we have

$$\begin{aligned} \tau'(x) &= 1 - \frac{1}{1 + e^x}, \\ \tau''(x) &= \frac{1}{(1 + e^x)^2} \cdot e^x = \frac{1}{\frac{1}{e^x} + e^x + 2} \leq \frac{1}{4}. \end{aligned}$$

As a result, the LHS of equation (5.2) satisfies:

$$LHS \leq \frac{1}{4} \langle \langle \boldsymbol{X}_i, \boldsymbol{\Phi} \rangle_F \boldsymbol{X}_i, \boldsymbol{\Phi} \rangle_F.$$

Also, for each sample i, we have

$$\begin{aligned} \langle \boldsymbol{X}_i, \boldsymbol{\Phi} \rangle_F =& \langle vec(\boldsymbol{X}_i), vec(\boldsymbol{\Phi}) \rangle_2 \\ =& vec(\boldsymbol{X}_i)^t vec(\boldsymbol{\Phi}), \end{aligned}$$

which leads to

$$\begin{aligned}\langle\langle \boldsymbol{X}_i, \boldsymbol{\Phi}\rangle_F \boldsymbol{X}_i, \boldsymbol{\Phi}\rangle_F =& \langle vec(\boldsymbol{X}_i)^t vec(\boldsymbol{\Phi}) vec(\boldsymbol{X}_i)^t vec(\boldsymbol{\Phi}) \\ =& vec(\boldsymbol{\Phi})^t vec(\boldsymbol{X}_i) vec(\boldsymbol{X}_i)^t vec(\boldsymbol{\Phi}).\end{aligned}$$

Summing all terms together with respect to i from 1 to n, we get

$$\sum_{i=1}^{n} \langle\langle \boldsymbol{X}_i, \boldsymbol{\Phi}\rangle_F \boldsymbol{X}_i, \boldsymbol{\Phi}\rangle_F = vec(\boldsymbol{\Phi})^t \boldsymbol{H} vec(\boldsymbol{\Phi}).$$

Since $\boldsymbol{H}$ is symmetric, denote the eigenvalues of $\boldsymbol{H}$ as $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{np}$. Then we should have

$$vec(\boldsymbol{\Phi})^t \boldsymbol{H} vec(\boldsymbol{\Phi}) \leq \lambda_1 \|vec(\boldsymbol{\Phi})\|_2^2 = \|\boldsymbol{H}\|_2 \|\boldsymbol{\Phi}\|_F^2.$$

As a result, equation (5.2) holds true.

Next, we will prove (5.1). Taking the difference between equation (2.17) and equation (3.5), we have

$$\begin{aligned}& G(\boldsymbol{B}, \boldsymbol{B}^*) - f(\boldsymbol{B}) \\ =& l(\boldsymbol{B}^*) - l(\boldsymbol{B}) + \langle \sum_{i=1}^{n} \{\tau'(\langle \boldsymbol{X}_i, \boldsymbol{B}^*\rangle_F) - y_i\} \boldsymbol{X}_i, \boldsymbol{B} - \boldsymbol{B}^*\rangle_F + \frac{1}{2}\|\boldsymbol{B} - \boldsymbol{B}^*\|_F^2.\end{aligned} \tag{5.3}$$

Taking Taylor expansion on $\tau(\langle \boldsymbol{X}, \boldsymbol{B}\rangle_F)$ at $\boldsymbol{B} = \boldsymbol{B}^*$, we have

$$\tau(\langle \boldsymbol{X}_i, \boldsymbol{B}\rangle_F) = \tau(\langle \boldsymbol{X}_i, \boldsymbol{B}^*\rangle_F) + \partial\tau(\langle \boldsymbol{X}_i, \boldsymbol{B}^*\rangle_F)^{\to \boldsymbol{B}-\boldsymbol{B}^*} + \frac{1}{2!}\partial^2\tau(\langle \boldsymbol{X}_i, \boldsymbol{B}^*\rangle_F)^{\to \boldsymbol{B}-\boldsymbol{B}^*} + O(\|\boldsymbol{B} - \boldsymbol{B}^*\|_F^3). \tag{5.4}$$

Since

$$\begin{cases} \partial\tau(\langle \boldsymbol{X}_i, \boldsymbol{B}^*\rangle_F)^{\to \boldsymbol{B}-\boldsymbol{B}^*} = \langle d'(\langle \boldsymbol{X}_i, \boldsymbol{B}^*\rangle_F) \boldsymbol{X}_i, \boldsymbol{B} - \boldsymbol{B}^*\rangle_F \\ \partial^2\tau(\langle \boldsymbol{X}_i, \boldsymbol{B}^*\rangle_F)^{\to \boldsymbol{B}-\boldsymbol{B}^*} = \langle \tau''(\langle \boldsymbol{X}_i, \boldsymbol{B}^*\rangle_F) \langle \boldsymbol{X}_i, \boldsymbol{B} - \boldsymbol{B}^*\rangle_F \boldsymbol{X}_i, \boldsymbol{B} - \boldsymbol{B}^*\rangle_F. \end{cases} \tag{5.5}$$

Substituting equation (5.5) into equation (5.4), we get

$$\tau(\langle \boldsymbol{X}_i, \boldsymbol{B}\rangle_F) - \tau(\langle \boldsymbol{X}_i, \boldsymbol{B}^*\rangle_F) = \langle \tau'(\langle \boldsymbol{X}_i, \boldsymbol{B}^*\rangle_F)\boldsymbol{X}_i, \boldsymbol{B} - \boldsymbol{B}^*\rangle_F + \frac{1}{2}\langle \tau''(\langle \boldsymbol{X}_i, \boldsymbol{B}^*\rangle_F)\langle \boldsymbol{X}_i, \boldsymbol{B} - \boldsymbol{B}^*\rangle_F \boldsymbol{X}_i, \boldsymbol{B} - \boldsymbol{B}^*\rangle_F + O(\|\boldsymbol{B} - \boldsymbol{B}^*\|_F^3).$$

Combing the above equation with equation (5.2) and (5.3), we get

$$\begin{aligned}
& G(\boldsymbol{B}, \boldsymbol{B}^*) - f(\boldsymbol{B}) \\
= & -\frac{1}{2}\sum_{i=1}^{n}\left\langle \tau''(\langle \boldsymbol{X}_i, \boldsymbol{B}^*\rangle_F)\langle \boldsymbol{X}_i, \boldsymbol{B} - \boldsymbol{B}^*\rangle_F \boldsymbol{X}_i, \boldsymbol{B} - \boldsymbol{B}^*\right\rangle_F + \frac{\rho}{2}\|\boldsymbol{B} - \boldsymbol{B}^*\|_F^2 \\
\geq & -\frac{1}{2}\frac{\|\boldsymbol{H}\|_2}{4}\|\boldsymbol{B} - \boldsymbol{B}^*\|_F^2 + \frac{\rho}{2}\|\boldsymbol{B} - \boldsymbol{B}^*\|_F^2 \\
\geq & \frac{\rho - \frac{\|\boldsymbol{H}\|_2}{4}}{2}\|\boldsymbol{B} - \boldsymbol{B}^*\|_F^2
\end{aligned}$$

leads to (5.1). The first equality in Theorem 2.3.1 holds true. The second inequality is a direct conclusion by definition of $\boldsymbol{B}^{[k+1]}$, and the last inequality is simply the construction of G. □

A.2 Duality Gap

We derive the dual problem of equation (3.5) in this section, and use logistic regression as an example to calculate the duality gap.

Using the notation of $\tilde{\boldsymbol{\Lambda}}$ and $\tilde{\boldsymbol{T}}$ in equation (2.21), (2.22), we have

$$f(\boldsymbol{B} \mid \boldsymbol{y}, \boldsymbol{X}) = l(\boldsymbol{B} \mid \boldsymbol{y}, \boldsymbol{X}) + \|\tilde{\boldsymbol{\Lambda}}\boldsymbol{T}\boldsymbol{B}\|_{2,1}. \tag{5.6}$$

For Logistic regression, we have $\tau(\eta_i) = \log(1 + e^{\eta_i})$ in equation (3.5). Let $l(\boldsymbol{\eta}) := \langle \boldsymbol{y}, \boldsymbol{\eta}\rangle_2 + \langle 1, \tau(\boldsymbol{\eta})\rangle_2$ and $P(\tilde{\boldsymbol{C}}) := \|\tilde{\boldsymbol{\Lambda}}\tilde{\boldsymbol{C}}\|_{2,1}$. Using equation (2.24), the optimization problem

in equation (**??**) is equivalent to

$$\min_{\tilde{C}} l(\boldsymbol{\eta}) + P(\tilde{\boldsymbol{C}}), \quad \eta_i = \langle \boldsymbol{X}_i, \boldsymbol{H}\tilde{\boldsymbol{C}} \rangle_F. \tag{5.7}$$

The Lagrangian is

$$\begin{aligned} L(\boldsymbol{\gamma}, \tilde{\boldsymbol{C}}) =& l(\boldsymbol{\eta}) + P(\tilde{\boldsymbol{C}}) + \sum_{i=1}^{n} \gamma_i (\eta_i - \langle \boldsymbol{X}_i, \boldsymbol{H}\tilde{\boldsymbol{C}} \rangle_F) \\ =& l(\boldsymbol{\eta}) + \langle \boldsymbol{\gamma}, \boldsymbol{\eta} \rangle_2 + P(\tilde{\boldsymbol{C}}) - \langle \sum_{i=1}^{n} \gamma_i \boldsymbol{H}^T \boldsymbol{X}_i, \tilde{\boldsymbol{C}} \rangle_F. \end{aligned}$$

$\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_n) \in \mathbb{R}^{n \times 1}$ is the dual variable. Since the equation is convex with equality constraints, strong duality holds true. The dual problem is

$$\max_{\boldsymbol{\gamma}} \min_{\tilde{\boldsymbol{C}}} L(\tilde{\boldsymbol{C}}, \boldsymbol{\eta}, \boldsymbol{\gamma}) = \max_{\boldsymbol{\gamma}} -l^*(\boldsymbol{\gamma}) - P^*(\sum_{i=1}^{n} \gamma_i \boldsymbol{H}^T \boldsymbol{X}_i) \tag{5.8}$$

where $l^*(\cdot)$ and $P^*(\cdot)$ are the Fenchel conjugate (Bauschke and Combettes 2011) of function $l(\cdot)$ and $P(\cdot)$. By definition of the conjugate function, for each $i = 1, \ldots, n$,

$$\begin{aligned} l^*(\gamma_i) =& \sup_{\eta_i} (\eta_i \gamma_i - l(\eta_i)) \\ =& (\gamma_i - y_i) \log \frac{\gamma_i - y_i}{1 - (\gamma_i - y_i)} + \log (1 - (\gamma_i - y_i)), \quad 0 \leq \gamma_i - y_i \leq 1. \end{aligned}$$

The supreme value of η_i is attained at $\gamma_i = y_i + \frac{1}{1+e^{-\eta_i}}$. As a result, we have

$$l^*(\boldsymbol{\gamma}) = \sum_{i=1}^{n} \{(\gamma_i - y_i) \log \frac{\gamma_i - y_i}{1 - (\gamma_i - y_i)} + \log (1 - (\gamma_i - y_i))\}, \quad 0 \leq \gamma_i - y_i \leq 1.$$

The conjugate function of $P(C)$ is

$$\begin{aligned} P^*(\Psi) &= \sup_{\tilde{C}}[\langle \Psi, \tilde{C}\rangle_F - \lambda\|\tilde{\Lambda}\tilde{C}\|_{2,1}] \\ &= \sum_{i=1}^{\tilde{l}} \sup_{\mathcal{S}_i}[\langle \tilde{\boldsymbol{c}}_i, \boldsymbol{\psi}_i\rangle_2 - \lambda w_i\|\tilde{\boldsymbol{c}}_i\|_2] \\ &= \sum_{i=1}^{\tilde{l}} \iota_{\mathcal{S}_i}(\boldsymbol{\psi}_l) \quad \mathcal{S}_i = \{z : \|z\|_2 \le \lambda w_i\} \end{aligned}$$

where $\boldsymbol{\psi}_i$, $\tilde{\boldsymbol{c}}_i$ is the i-th row of Ψ, $\tilde{C}$, and $\iota_{\mathcal{S}}(\cdot)$ is the indicator function in set $\mathcal{S}$. As a result, the dual problem is equivalent to

$$\max_{\boldsymbol{\gamma}} \sum_{i=1}^{n} (\gamma_i - y_i)\log\frac{\gamma_i - y_i}{1-(\gamma_i - y_i)} + \log(1-(\gamma_i - y_i)), \quad s.t. \quad \|\boldsymbol{\psi}_i\|_2 \le \lambda w_i$$

where $\Psi = \sum_{i=1}^{n} \gamma_i \boldsymbol{H}^T \boldsymbol{X}_i = \boldsymbol{H}^T \mathrm{diag}(\boldsymbol{\gamma})\boldsymbol{X}$. Denote

Prime: $f(\tilde{C}) = l(\boldsymbol{\eta}) + P(\tilde{C})$;

Dual: $D(\boldsymbol{y}, \boldsymbol{\gamma}) = \max_{\boldsymbol{\gamma}} \sum_{i=1}^{n} (\gamma_i - y_i)\log\frac{\gamma_i - y_i}{1-(\gamma_i - y_i)} + \log(1-(\gamma_i - y_i))$.

The duality gap between the primal problem and the dual problem should be $|f(\tilde{C}) - D(\boldsymbol{y}, \boldsymbol{\gamma})|$. As the algorithm converges, the duality gap should decrease to 0.

A.3 Algorithm Developed Directly from Equation (3.5)

ADMM (Boyd, Parikh, and Chu 2011; Gabay and Mercier 1975; Glowinski and Marroco 1975) is an operator splitting method motivated from a variant of the augmented Lagrangian method (Hestenes 1969; Wright and Nocedal 1999; Rockafellar 1973). It adds a quadratic augmented term to the original Lagrangian to make it strongly convex. We can use it to

solve equation 3.5 directly. We introduce $\boldsymbol{C} = \boldsymbol{TB}$, then equation 3.5 is equivalent to

$$\min_{\boldsymbol{B}} l(\boldsymbol{B}|\boldsymbol{y}, \boldsymbol{X}) + P(\boldsymbol{C}), \quad s.t.\ \boldsymbol{TB} = \boldsymbol{C}.$$

We add an augmented term with $\nu > 0$ to the loss function and get

$$\min_{\boldsymbol{B},\boldsymbol{C}} l(\boldsymbol{B}|\boldsymbol{y}, \boldsymbol{X}) + P(\boldsymbol{C}) + \frac{\nu}{2}\|\boldsymbol{TB} - \boldsymbol{C}\|_F^2, \quad s.t.\ \boldsymbol{TB} = \boldsymbol{C}.$$

The Lagrangian is

$$L_\nu(\boldsymbol{B}|\boldsymbol{y}, \boldsymbol{X}) = l(\boldsymbol{B}|\boldsymbol{y}, \boldsymbol{X}) + P(\boldsymbol{C}) + \frac{\nu}{2}\|\boldsymbol{TB} - \boldsymbol{C}\|_F^2 + \langle \boldsymbol{\Gamma}, \boldsymbol{TB} - \boldsymbol{C}\rangle.$$

ADMM iteratively update $\boldsymbol{B}$ and $\boldsymbol{C}$ separately. Given $\boldsymbol{C}^{[k]}$ and $\boldsymbol{\Gamma}^{[k]}$ at step k,

$$\begin{aligned}\boldsymbol{B}^{[k+1]} &= \arg\min_{\boldsymbol{B}} L_\nu(\boldsymbol{B}, \boldsymbol{C}^{[k]}, \boldsymbol{\Gamma}^{[k]}) \\ &= \arg\min_{\boldsymbol{B}} -\langle \boldsymbol{y}, \boldsymbol{\eta}\rangle + \langle 1, b(\boldsymbol{\eta})\rangle + \frac{\nu}{2}\|\boldsymbol{TB} - \boldsymbol{C}^{[k]} + \frac{1}{\nu}\boldsymbol{\Gamma}^{[k]}\|_F^2.\end{aligned}$$

This is a un-constrained convex optimization problem. Thus it can be solved using the Matlab package MinFunc. On the other hand, once we get newly updated $\boldsymbol{B}$ and $\boldsymbol{\Gamma}$, Updating $\boldsymbol{C}$ is equivalent to

$$\boldsymbol{C} = \arg\min_{\boldsymbol{C}} \frac{1}{2}\|\boldsymbol{C} - (\boldsymbol{TB} - \frac{1}{\nu}\boldsymbol{\Gamma})\|_F^2 + \frac{1}{\nu}\|\boldsymbol{\Lambda C}\|_{2,2}.$$

The above equation for $\boldsymbol{C}$ is separable for each row, and each row of $\boldsymbol{C}$ can be solved by the proximal mapping. To be more specific, for $\boldsymbol{C} = (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_l)^T$, $l = \frac{n(n-1)}{2}$, we have

$$\boldsymbol{c}_i = \mathbf{prox}_{\sigma_i\|\cdot\|_2}(\boldsymbol{t}_i\boldsymbol{B} - \frac{1}{\nu}\boldsymbol{\gamma}_i), \quad i = 1, \ldots, l,$$

where $\boldsymbol{t}_i$, $\boldsymbol{\gamma}_i$ is the ith row of $\boldsymbol{T}$ and Γ respectively, and $\sigma_i = \lambda x w_i / v$. Finally, we update the dual variable Γ according to the dual update:

$$\Gamma^{k+1} = \Gamma^k + v(\boldsymbol{TB} - \boldsymbol{C}). \tag{5.9}$$

In summary, the algorithm can be described as below

Algorithm 6 Algorithm to solve equation (3.5)

Initialize $\Gamma^{[0]}$ and $\boldsymbol{C}^{[0]}$

for $k = 0, 1, 2 \ldots$ **do**

$\boldsymbol{B}^{[k+1]} = \arg\min_{\boldsymbol{B}} -\langle \boldsymbol{y}, \boldsymbol{\eta} \rangle + \langle 1, b(\boldsymbol{\eta}) \rangle + \frac{v}{2} \| \boldsymbol{TB} - \boldsymbol{C}^{[k]} + \frac{1}{v} \Gamma^{[k]} \|_F^2$.

for $i = 1, \ldots l$ **do**

$\boldsymbol{c}_l^{[k+1]} = prox_{\sigma_l}(\boldsymbol{b}_{i1}^{[k+1]} - \boldsymbol{b}_{i2}^{[k+1]} - v^{-1} \boldsymbol{\gamma}_i^{[k]})$, where $\sigma_i = \lambda w_i / v$.

$\boldsymbol{\gamma}_l^{[k+1]} = \boldsymbol{\gamma}_i^{[k-1]} + v(\boldsymbol{c}_i^{[k]} - \boldsymbol{b}_{i1}^{[k+1]} + \boldsymbol{b}_{i2}^{[k+1]})$.

A.4 Proof of Theorem 2.4.1

We first introduce some notations to prove the theorem. Let $V_j(n, d)$ denote the volume of a Hamming ball of radius $d \leq J$ in $\{0, 1, \ldots, J-1\}$. i.e.,

$$V_J(n, d) = \sum_{i=0}^{\lfloor d \rfloor} \binom{n}{i} (b-1)^i.$$

Given $J \geq 2$, $x \in [0, 1]$, define the J-ary entropy function

$$h_J(x) = x \log_J(J-1) - x \log_J x - (1-x) \log_J(1-x),$$

and $h(x)$ is the Shannon entropy function. The following result is well known, see, e.g. Van Lint and Geer (2012).

Lemma 5.0.1. *Let $J \geq 2$, $d \in [0, n(1-1/J)]$, and $J, d \in \mathbb{Z}$. Then*

$$V_J(n,d) \leq J^{h_J(\theta)n},\ V_J(n,d) \geq \binom{n}{d}(J-1)^d \geq J^{h_J(\theta)n}\exp(-c\log n - c'),$$

where $\theta = d/n$, and c, c' are positive constants.

The next lemma is essentially the Gilbert-Varshamov bound for J-ary codes, adapted for our purposes.

Lemma 5.0.2. *Let $\Omega = \{\boldsymbol{a} = (a_1, \ldots, a_n)^T, a_j \in \mathcal{A}\}$, where $\mathcal{A}$ is a set with cardinality $|\mathcal{A}| = J$, $(2 \leq J \leq n)$. Then there exists a subset $\{\boldsymbol{a}^0, \ldots, \boldsymbol{a}^M\} \subset \Omega$ such that $\boldsymbol{a}^0 \in \mathcal{A}^n$ is arbitrarily chosen, and*

$$\log M \geq \log(J^n/V_J(n, \lceil d\rceil - 1) - 1) \geq c_1 n \log J,$$
$$\rho(\boldsymbol{a}^j, \boldsymbol{a}^k) \geq c_2 n,\ \forall 0 \leq j < k \leq M,$$

where $\rho(\boldsymbol{a}, \boldsymbol{a}') = \sum_{i=1}^n 1_{a_j \neq a_{j'}}$, and c_1, c_2 are universal positive constants.

Proof. Let $d = c_2 n$. Given any $\boldsymbol{a} \in \Omega$, the number of elements in $\{\boldsymbol{b} \in \Omega\colon \rho(\boldsymbol{b}, \boldsymbol{a}) \leq l\} =: B_J(\boldsymbol{a}; l)$ is no more than $\sum_{i=0}^{l}\binom{n}{i}(J-1)^i$ which is just $V_J(n,l)$.

Consider the following procedure to partition Ω. Let $\boldsymbol{a}^0 \in \mathcal{A}^n$ be arbitrarily chosen and $\Omega^0 = \Omega$. Given $\boldsymbol{a}^t \in \Omega^t$ $(t \geq 0)$, construct $L^t = \{\boldsymbol{a} \in B_J(\boldsymbol{a}^t; \lceil d\rceil - 1)\} \cap \Omega^t$ and $\Omega^{t+1} = \Omega^t \setminus L^t$, and choose an arbitrary $\boldsymbol{a}^{t+1} \in \Omega^{t+1}$. Repeat the process until $\Omega^{M+1} = \emptyset$. Then L^t $(0 \leq t \leq M)$ form a partition of Ω, and so

$$(M+1)V_J(n, \lceil d\rceil - 1) \geq J^n,$$

By Lemma 5.0.1, for $d/n \leq 1 - 1/J$ or $c_2 \leq 1 - 1/J$

$$M + 1 \geq J^{(1-h_J(c_2))n}.$$

It is not difficult to show that with a small enough c_2, $J^{(1-h_J(c_2))n} \geq 1 + J^{c_1 n}$ holds for all $n \geq J \geq 2$ and some constant $c_1 > 0$. The conclusion follows. □

The next lemma characterizes the relationship between Kullback-Leibler divergence (Kullback and Leibler 1951) and generalized Bregman divergence.

Lemma 5.0.3. *For $P_{\bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}}$ defined above, the Kullback-Leibler (K-L) divergence of $P_{\bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}_1}$ from $P_{\bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}_2}$ satisfies*

$$KL(P_{\bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}_1} \| P_{\bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}_2}) = \Delta l_0(\bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}_1, \bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}_2) \leq \frac{1}{4}\|\bar{\boldsymbol{X}}\|_2^2 \|\bar{\boldsymbol{\beta}}_1 - \boldsymbol{\beta}_2\|_2^2/2. \tag{5.10}$$

Furthermore, we have $\|\bar{\boldsymbol{X}}\|_2 = \|\boldsymbol{X}\|_{2,\infty}$.

Proof. According to the definition of K-L divergence, we get

$$\begin{aligned}
&\mathrm{KL}(P_{\bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}_1} \| P_{\bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}_2}) \\
=&\int P_{\bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}_1}(\boldsymbol{y})[\log(P_{\bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}_1}(\boldsymbol{y}) - \log(P_{\bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}_2}(\boldsymbol{y})]d\boldsymbol{y} \\
=&\int P_{\bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}_1}(\boldsymbol{y})[\langle \boldsymbol{y}, \boldsymbol{\eta}_2 - \boldsymbol{\eta}_1\rangle + \langle 1, \tau(\boldsymbol{\eta}_2) - \tau(\boldsymbol{\eta}_1)\rangle]d\boldsymbol{y} \\
=&\langle 1, \tau(\boldsymbol{\eta}_2) - \tau(\boldsymbol{\eta}_1)\rangle_2 - \langle \mathbb{E}_{P_{\bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}_1}}\{Y\}, \boldsymbol{\eta}_2 - \boldsymbol{\eta}_1\rangle_2 \\
=&\langle 1, \tau(\boldsymbol{\eta}_2) - \tau(\boldsymbol{\eta}_1)\rangle_2 - \langle \nabla l_0(\boldsymbol{\eta}_1), \boldsymbol{\eta}_2 - \boldsymbol{\eta}_1\rangle \\
=&\Delta_{l_0}(\boldsymbol{\eta}_2, \boldsymbol{\eta}_1).
\end{aligned} \tag{5.11}$$

Thus, the first inequality in (5.10) holds true. On the other hand

$$\Delta_{l_0}(\boldsymbol{\eta}_2, \boldsymbol{\eta}_1) = l_0(\boldsymbol{\eta}_2) - l_0(\boldsymbol{\eta}_1) - \langle \nabla l_0(\boldsymbol{\eta}_1), \boldsymbol{\eta}_2 - \boldsymbol{\eta}_1\rangle. \tag{5.12}$$

The Taylor expansion of $l_0(\boldsymbol{\eta})$ at $\boldsymbol{\eta} = \boldsymbol{\eta}_1$ is

$$l_0(\boldsymbol{\eta}) = l_0(\boldsymbol{\eta}_1) + \nabla l_0(\boldsymbol{\eta}_1)(\boldsymbol{\eta} - \boldsymbol{\eta}_1) + \frac{1}{2}(\boldsymbol{\eta} - \boldsymbol{\eta}_1)^T \nabla^2 l_0(\boldsymbol{\eta}_1)(\boldsymbol{\eta} - \boldsymbol{\eta}_1) + o(\|\boldsymbol{\eta} - \boldsymbol{\eta}_1\|_2^2),$$

where

$$\begin{cases} \nabla l_0(\boldsymbol{\eta}) = -\boldsymbol{y}^T + \tau'(\boldsymbol{\eta})^T, \\ \nabla^2 l_0(\boldsymbol{\eta}) = diag(\tau''(\boldsymbol{\eta})) \end{cases}$$

Since $\tau''(\eta_i) \leq \frac{1}{4}$ $(1 \leq i \leq n)$, let $\boldsymbol{\eta} = \boldsymbol{\eta}_2$ and ignoring the higher order term when $\boldsymbol{\eta}_2 \to \boldsymbol{\eta}_1$, we obtain

$$l_0(\boldsymbol{\eta}_2) - l_0(\boldsymbol{\eta}_1) - \nabla l_0(\boldsymbol{\eta}_1)(\boldsymbol{\eta}_2 - \boldsymbol{\eta}_1) \leq \tfrac{1}{4}\|\boldsymbol{\eta}_2 - \boldsymbol{\eta}_1\|_2^2/2 \leq \tfrac{1}{4}\|\bar{\boldsymbol{X}}\|_2^2\|\bar{\boldsymbol{\beta}}_2 - \bar{\boldsymbol{\beta}}_1\|_2^2/2. \quad (5.13)$$

Combining equation (5.12) and equation (5.13), the inequality in equation (5.10) holds.

Furthermore, $\|\bar{\boldsymbol{X}}\|_2 = \sqrt{\lambda_{max}(\bar{\boldsymbol{X}}^T\bar{\boldsymbol{X}})}$, namely, the 2 norm of $\bar{\boldsymbol{X}}$ is the largest eigenvalue of $\bar{\boldsymbol{X}}^T\bar{\boldsymbol{X}}$. Since $\bar{\boldsymbol{X}}^T\bar{\boldsymbol{X}} = diag(\boldsymbol{x}_i\boldsymbol{x}_i^T)$, and the largest eigenvalue of $\boldsymbol{x}_i\boldsymbol{x}_i^T$ is $\boldsymbol{x}_i^T\boldsymbol{x}_i$. Thus,

$$\lambda_{max}(\bar{\boldsymbol{X}}^T\bar{\boldsymbol{X}}) = \max_{1\leq i\leq n}(\boldsymbol{x}_i^T\boldsymbol{x}_i) = \|\boldsymbol{X}\|_{2,\infty}.$$

The conclusion follows. □

To Prove the theorem, we consider the following two cases:

Case (i). $n\log J \geq pJ$:

Let b, ζ be integers satisfying

$$V_b(p,\zeta) = \sum_{j=0}^{\zeta}\binom{p}{j}(b-1)^j \geq J, b \geq 2, 1 \leq \zeta \leq J.$$

Thus we can make

$$C = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_J : \boldsymbol{c}_j \in \mathbb{R}^p, c_{j,k} \in \{0,1,\ldots,b-1\}, \|\boldsymbol{c}_j\|_0 \leq \zeta, \forall 1 \leq j \leq J\}$$

with $|C| = J$ and $0 \in C$. Next, construct

$$\mathcal{B}^1(J) := \{\boldsymbol{B} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_n)^T : \boldsymbol{\beta}_j \text{ satisfies } \boldsymbol{\beta}_j = 0 \text{ or } \frac{\boldsymbol{\beta}_j}{\gamma R} \in C \text{ for any } 1 \le j \le n\},$$

where $\gamma > 0$ is a small constant to be chosen later, and $R = \frac{\sqrt{\log J}}{K}$. We can set $K = \frac{1}{4}\|\bar{\boldsymbol{X}}\|_2^2$ Clearly, $\mathcal{B}^1(J) \subset \mathcal{S}(J)$.

Define the (vectorized) Hamming distance by

$$\rho(\boldsymbol{B}_1, \boldsymbol{B}_2) = \sum_{j=1}^{n} 1_{\boldsymbol{B}_1[j,:] \neq \boldsymbol{B}_2[j,:]},$$

where $\boldsymbol{B}_i[j,:]$ is the j-th row of matrix $\boldsymbol{B}_i$ $(i = 1, 2)$. From Lemma 5.0.2, there exists a subset $\mathcal{B}^{10}(J) \subset \mathcal{B}^1(J)$ such that $0 \in \mathcal{B}^{10}(J)$ and

$$\log(\left|\mathcal{B}^{10}(J)\right| - 1) \ge \log\left(J^n/(V_J(n, \lceil d \rceil - 1) - 1)\right) \ge c_1 n \log J \text{ and}$$
$$d = \rho(\boldsymbol{B}_1, \boldsymbol{B}_2) \ge c_2 n, \forall \boldsymbol{B}_1, \boldsymbol{B}_2 \in \mathcal{B}^{10}, \boldsymbol{B}_1 \neq \boldsymbol{B}_2$$

for some universal constants $c_1, c_2 > 0$. Hence

$$\|\boldsymbol{B}_1 - \boldsymbol{B}_2\|_F^2 \ge \gamma^2 R^2 \rho(\boldsymbol{B}_1, \boldsymbol{B}_2) \cdot 1 \ge c_2 \gamma^2 R^2 n = c_2 \gamma^2 \frac{n \log J}{K}, \tag{5.14}$$

for any $\boldsymbol{B}_1, \boldsymbol{B}_2 \in \mathcal{B}^{10}$, $\boldsymbol{B}_1 \neq \boldsymbol{B}_2$ where c_2 is a positive constant.

On the other hand, using Lemma 5.0.3, we have

$$\mathrm{KL}(P_{\bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}_1} \| P_{\bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}2}) \le \frac{1}{4}\|\bar{\boldsymbol{X}}\|_2^2 \|\boldsymbol{B}_1 - \boldsymbol{B}_2\|_F^2/2.$$

Thus, for any $\boldsymbol{B} \in \mathcal{B}^{10}$

$$\begin{aligned} &\mathrm{KL}(P_{\bar{X}0} \| P_{\bar{X}\bar{\boldsymbol{\beta}}}) \\ \leq &\frac{1}{4}\|\bar{\boldsymbol{X}}\|_2^2 \|\boldsymbol{B}\|_F^2/2 \leq \frac{1}{4}\|\bar{\boldsymbol{X}}\|_2^2 \gamma^2 R^2 \rho(0, \boldsymbol{B})/2 \leq \frac{1}{8}\|\bar{\boldsymbol{X}}\|_2^2 \gamma^2 R^2 n = \frac{1}{2}\gamma^2 n \log J. \end{aligned} \tag{5.15}$$

Therefore,

$$\frac{1}{|\mathcal{B}^{10}| - 1} \sum_{\boldsymbol{B} \in \mathcal{B}^{10} \setminus \{0\}} \mathrm{KL}(P_{\bar{X}0} \| P_{\bar{X}\bar{\boldsymbol{\beta}}}) \leq \gamma^2 \frac{n \log J}{2} \tag{5.16}$$

Now, combining (5.14) and (5.16) and choosing a sufficiently small value for γ, we can apply Theorem 2.7 of Tsybakov (2008) to get the lower bound of $(n - J) \log J/K$.

Case (ii). $n \log J < pJ$:

Consider a signal subclass

$$\mathcal{B}^2(J) = \{\boldsymbol{B} = [b_{jk}] : b_{jk} = 0 \text{ or } \gamma R \text{ if } 1 \leq j \leq J - 1, 1 \leq k \leq P \text{ and } b_{jk} = 0 \text{otherwise}\},$$

where $R = \sqrt{\frac{1}{K}}$ with $K = \frac{1}{4}\|\bar{\boldsymbol{X}}\|_2^2$ and $\gamma > 0$ is a small constant to be chosen later. Clearly, $|\mathcal{B}^2(J)| = 2^{pJ}$, $\mathcal{B}^2(J) \subset \mathcal{S}(J)$. In this case, we let $\rho(\boldsymbol{B}_1, \boldsymbol{B}_2) = \|\bar{\boldsymbol{\beta}}_1 - \bar{\boldsymbol{\beta}}_2\|_0$ be the Hamming distance. By Lemma 5.0.2 and $Jp \geq 2$, there exists a subset $\mathcal{B}^{20}(J) \subset \mathcal{B}^2(J)$ such that $0 \in \mathcal{B}^{20}$,

$$\log(|\mathcal{B}^{20}(J)| - 1) \geq c_1 Jp$$

and

$$\rho(\boldsymbol{B}_1, \boldsymbol{B}_2) \geq c_2 Jp, \forall \boldsymbol{B}_1, \boldsymbol{B}_2 \in \mathcal{B}^{20}, \boldsymbol{B}_1 \neq \boldsymbol{B}_2$$

for some universal constants $c_1, c_2 > 0$.

$$\|\boldsymbol{B}_1 - \boldsymbol{B}_2\|_2^2 \geq \gamma^2 R^2 \rho(\boldsymbol{B}_1, \boldsymbol{B}_2) \geq \frac{c_1 \gamma^2 Jp}{K} = 4c_1 \gamma^2 \frac{Jp}{\|\bar{\boldsymbol{X}}\|_2^2} \tag{5.17}$$

Furthermore, for any $\boldsymbol{B} \in \mathcal{B}^{20}(J)$, using Lemma 5.0.3 again, we have

$$\mathrm{KL}(P_{\bar{X}0} \| P_{\bar{X}\bar{\boldsymbol{\beta}}}) \leq frac14 \|\bar{\boldsymbol{X}}\|_2^2 \gamma^2 \frac{1}{K} \rho(0, \boldsymbol{B})/2 = \frac{1}{8} \|\bar{\boldsymbol{X}}\|_2^2 \gamma^2 Jp = \frac{1}{2} \gamma^2 Jp. \tag{5.18}$$

The afterwards treatment follows the same lines as in case (i) and the details are omitted.

The above two cases concludes the minimax lower bound rate of $[Jp + n \log J]/K$ with K can be chosen as $\|\boldsymbol{X}\|_{2,\infty}$. This is a conservative bound. In fact, it can be smaller because $\boldsymbol{\beta}_i$s are subject to equi-sparsity constraints. But it suffices for our purpose.

A.5 Proof of Theorem 2.4.2

In order to prove Theorem 2.4.2, we first introduce two lemmas. The first lemma introduce the lower and upper bound of the penalty.

Lemma 5.0.4. *Let the group size of each group be g_i ($1 \leq i \leq J$). Then $P(\bar{\boldsymbol{\beta}}) = \sum_{1 \leq i < i' \leq J} g_i g_{i'}$. Then the upper bound of $P(\bar{\boldsymbol{\beta}})$ is of the order $\frac{n^2}{2}(1 - \frac{1}{J})$, and the lower bound fo $P(\bar{\boldsymbol{\beta}})$ is of the order $\frac{1}{2}(J-1)(2n-J)$.*

Proof. We first explore the upper bound. Denote $f^J(n) = \sum_{1 \leq i < i' \leq J} g_i g_{i'}$. We want to prove that $f^J(n)$ obtains its maximum when

$$g_i = \begin{cases} \lfloor \frac{n}{J} \rfloor & \text{if } \frac{2n}{J} < \lfloor \frac{n}{J} \rfloor + \lceil \frac{n}{J} \rceil, \\ \lceil \frac{n}{J} \rceil & \text{otherwise.} \end{cases}$$

for $1 \leq i \leq J-1$ and $g_J = n - (J-1)g_i$. Specifically, when $J \mid n$, we have $g_i = \frac{n}{J}$ for $1 \leq i \leq J$. When $J - 2$, its trivial to verify the statement holds. Assuming for given J, the assumption is true as well, then for $j + 1$, we have

$$f^{J+1}(n) = \sum_{1 \leq i < i' \leq J} g_i g_{i'} + g_{J+1} \sum_{i=1}^{J} g_i. \tag{5.19}$$

Using the assumption, the first term in the above equation obtains its maximum when $g_1 = \ldots = g_J = x$ where $x \in \mathbb{N}^+$. Thus, we get

$$\begin{cases} f^{J+1}(n) = \binom{J}{2}x^2 + g_{J+1} \cdot Jx, \\ Jx + g_{J+1} = n. \end{cases} \tag{5.20}$$

As a result,

$$f^{J+1}(n) = \binom{J}{2}x^2 + (n - Jx)Jx = -\frac{J^2 + J}{2}(x - \frac{n}{J+1}) + \frac{Jn^2}{2(J+1)}. \tag{5.21}$$

x obtains its maximum at $x = \frac{n}{J+1}$ if $n \mid J+1$, or the closest integer to $\frac{n}{J+1}$ if $n \nmid J+1$. Thus, using mathematical induction, the statement hold true for any $J \geq 2$. The maximum of $f^J(n)$ obtains at $g_i = \frac{p}{J}$ on average. The corresponding value is $\frac{n^2}{2}\frac{J-1}{J} \sim \frac{n^2}{2}$.

For the lower bound, it is easy to verify that

$$\begin{aligned} f^J(n) =& f^{J-1}(n - g_J) + (n - g_J)g_J \\ =& \cdots \\ =& (n - g_J)g_J + \ldots + (n - g_J - \cdots - g_1)g_1 \\ =& n^2 - [g_J(g_J + \cdots + g_1) + \cdots + g_1^2]. \end{aligned} \tag{5.22}$$

Let $S_i = \sum_{j=1}^{i} g_j$, then $g_i = S_i - S_{i-1}$ satisfying $S_1 < S_2 < \ldots < S_J = n$. Thus, we have $f^J(n) = n^2 - \sum_{j=1}^{J} S_j g_j$. Denote $H_J(S_1, \ldots, S_J) = S_1^2 + \sum_{j=1}^{J} S_j g_j$. Since

$$\sum_{j=1}^{J} S_j g_j = S_1^2 + \ldots + S_J(S_J - S_{J-1}) = S_1^2 + \sum_{j=2}^{J} S_j(S_j - S_{j-1}),$$

we want to prove $H_J(S_1, \ldots, S_J)$ obtains its maximum when $S_j = j$ $(1 \leq j \leq J-1)$ given S_J. When $J = 2$,

$$H_2(S_1, S_2) = S_1^2 + S_2(S_2 - S_1) = S_1^2 + n^2 - nS_1,$$

the statement hold naturally. When $J > 2$, assume the statement holds, then for $H_{J+1}(S_1, \ldots, S_{J+1})$, we get

$$H_{J+1}(S_1, \ldots, S_{J+1}) = H_J(S_1, \ldots, S_J) + S_{J+1}(S_{J+1} - S_J).$$

Given any $S_J = S^* \geq J$, according to the assumption, the first term in the above equation obtains its maximum when $S_j = j$ $(1 \leq j \leq J-1)$, and for the second term, it is decreasing as S_J increases. Thus, we want S_J as small as possible. Namely, $S_J = J$. As a result, the statement also holds for $J+1$. Using mathematical induction, the statement holds for all $J \geq 2$. We obtain the maximum of $H_J(S_1, \ldots, S_J)$, which is also the minimum of $f^J(n)$ at $g_j = 1$ $(1 \leq j \leq J-1)$ and $g_J = n - (J-1)$. The corresponding value of $f^J(o)$ is $\frac{1}{2}(J-1)(2n-J)$, which concludes the lemma. □

Next, we introduce the notion of noise. For the loss function $l_0(\boldsymbol{\eta})$ defined in equation (2.36), we define the effective noise associated with the statistical truth ($\boldsymbol{\eta}^* = \bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}^*$) by

$$\boldsymbol{\epsilon} = -\nabla l_0(\boldsymbol{\eta}^*). \tag{5.23}$$

Under some regularity condition to permit the exchange of the gradient and expectation, having a zero mean noise implies the expected loss vanishes at the statistical truth. In our case, the noise is $\boldsymbol{\epsilon} = \boldsymbol{Y} - \mathbb{E}(\boldsymbol{Y})$. We assume $\boldsymbol{\epsilon}$ to be a sub-Gaussian random vector with mean zero and scale bounded by σ. Lemma 5.0.5 shows that the noise satisfies the following relationship:

Lemma 5.0.5.

$$\mathbb{E}[\langle \boldsymbol{\epsilon}, \bar{\boldsymbol{X}}\hat{\bar{\boldsymbol{\beta}}} - \bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}^* \rangle] \leq \left(\frac{2}{a} + \frac{2}{a'}\right) \frac{\|\bar{\boldsymbol{X}}\hat{\bar{\boldsymbol{\beta}}} - \bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}^*\|^2}{2} + bL\sigma^2(P(\hat{\bar{\boldsymbol{\beta}}}) + P(\bar{\boldsymbol{\beta}}^*)) + ca'\sigma^2 \tag{5.24}$$

for any a, b, $a' > 0$, $4b > a$, where c, L are universal constants.

The detailed proof of the lemma could be found in She and Zhang (2020).

Since $\hat{\bar{\boldsymbol{\beta}}}$ is the optimal solution of (2.39), we have

$$l_0(\bar{\boldsymbol{X}}\hat{\bar{\boldsymbol{\beta}}}) + \frac{\lambda^2}{2}P(\hat{\bar{\boldsymbol{\beta}}}) \leq l_0(\bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}^*) + \frac{\lambda^2}{2}P(\bar{\boldsymbol{\beta}}^*). \tag{5.25}$$

Combing the definition of Bregman divergence in equation (2.37), we get

$$\Delta_{l_0}(\hat{\boldsymbol{\eta}}, \boldsymbol{\eta}^*) + \frac{\lambda^2}{2}P(\hat{\bar{\boldsymbol{\beta}}}) \leq \frac{\lambda^2}{2}P(\bar{\boldsymbol{\beta}}^*) + \langle \boldsymbol{\epsilon}, \hat{\boldsymbol{\eta}} - \boldsymbol{\eta}^* \rangle.$$

Taking expectation on both sides and combining Lemma 5.0.5, we get

$$\begin{aligned}
&\mathbb{E}[\Delta_{l_0}(\hat{\boldsymbol{\eta}}, \boldsymbol{\eta}^*)] + \frac{\lambda^2}{2}P(\hat{\bar{\boldsymbol{\beta}}}) \\
\leq &\frac{\lambda^2}{2}P(\bar{\boldsymbol{\beta}}^*) + \mathbb{E}\{\langle \boldsymbol{\epsilon}, \hat{\boldsymbol{\eta}} - \boldsymbol{\eta}^* \rangle\} \\
\leq &\frac{\lambda^2}{2}P(\bar{\boldsymbol{\beta}}^*) + \mathbb{E}\left\{\left(\frac{2}{a} + \frac{2}{a'}\right)\frac{\|\bar{\boldsymbol{X}}\hat{\bar{\boldsymbol{\beta}}} - \bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}^*\|_2}{2}\right\} + bL\sigma^2(P(\hat{\bar{\boldsymbol{\beta}}}) + P(\bar{\boldsymbol{\beta}})) + ca'\sigma^2.
\end{aligned} \tag{5.26}$$

Specifically, we can choose $a = a' = \frac{8}{\mu}$, $b = \frac{3}{u}$. Then using the restricted condition in inequation (2.38), we get

$$\begin{aligned}
\mathbb{E}\left[\frac{\mu}{2}\mathrm{D}_2(\bar{\boldsymbol{X}}\hat{\bar{\boldsymbol{\beta}}}, \bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}^*)\right] - K\lambda_0^2 P(\hat{\bar{\boldsymbol{\beta}}}) + \frac{\lambda^2}{2}P(\hat{\bar{\boldsymbol{\beta}}}) - \frac{3L\sigma^2}{u}P(\hat{\bar{\boldsymbol{\beta}}}) \leq K\lambda_2^2 P(\bar{\boldsymbol{\beta}}^*) + \\
\frac{\lambda^2}{2}P(\bar{\boldsymbol{\beta}}^*) + \frac{3L\sigma^2}{u}P(\bar{\boldsymbol{\beta}}^*) + ca'\sigma^2.
\end{aligned} \tag{5.27}$$

Adding Rp on both sides of the inequation where $R = \frac{3A^2}{4}(K \vee \frac{1}{\mu})$. Since we can choose sufficiently large A such that

$$\begin{cases}
\frac{\lambda^2}{2} - K\lambda_0^2 + \frac{3L\sigma^2}{\mu} = \frac{A^2}{2}(K \vee \frac{1}{\mu})\lambda_0^2 - K\lambda_0^2 - \frac{3L\sigma^2}{\mu} \geq 0, \\
\frac{c\sigma^2}{8\mu} - \frac{3A^2}{4}\left(k \vee \frac{1}{\mu}\right) \leq 0.
\end{cases}$$

Thus, we have

$$(\frac{\lambda^2}{2} - K\lambda_0^2 - \frac{3L\sigma^2}{\mu})P(\hat{\bar{\boldsymbol{\beta}}}) \geq 0.$$

Inequation (5.27) becomes

$$\mathbb{E}\left[\frac{\mu}{2}\mathrm{D}_2(\bar{\boldsymbol{X}}\hat{\bar{\boldsymbol{\beta}}}, \bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}^*)\right] \leq \left(K\lambda_0^2 + \frac{\lambda^2}{2} + \frac{3L\sigma^2}{\mu}\right)P(\bar{\boldsymbol{\beta}}^*) + Rp + (ca'\sigma^2 - Rp). \tag{5.28}$$

Similarly, we have

$$K\lambda_0^2 + \frac{\lambda^2}{2} + \frac{3L\sigma^2}{\mu} \leq \frac{3A^2}{2}(K \vee \frac{1}{\mu})\lambda_0^2.$$

Substituting it into inequation (5.28) and using Lemma 5.0.4, we get

$$\mathbb{E}\left[\frac{\mu}{2}\mathrm{D}_2(\bar{\boldsymbol{X}}\hat{\bar{\boldsymbol{\beta}}}, \bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}^*)\right] \leq \frac{3A^2}{2}\left(K \vee \frac{1}{\mu}\right)[P(\bar{\boldsymbol{\beta}}^*) + \frac{p}{2}].$$

Thus, we have

$$\begin{aligned}\mathbb{E}\left[\mathrm{D}_2(\bar{\boldsymbol{X}}\hat{\bar{\boldsymbol{\beta}}}, \bar{\boldsymbol{X}}\bar{\boldsymbol{\beta}}^*)\right] &\leq C\left(\frac{K\mu \vee 1}{\mu^2}\right)[P(\bar{\boldsymbol{\beta}}^*) + p] \\ &\leq C\left(\frac{K\mu \vee 1}{\mu^2}\right)[p + (p+n)(n - \frac{n}{J^*})].\end{aligned} \tag{5.29}$$

A.6 Proof of Theorem 4.2.2

Proof. From the definition of $T_k(\boldsymbol{s}_i)$, we have

$$T_k(\boldsymbol{s}_i) \leq \max_{0\leq j<n_i^*} |F_k^{(i)}(d_j) - \tilde{F}_k^{(i)}(d_j)|.$$

As a result, we only need to find the upper bound of

$$\mathcal{F} := \max_{0\leq j<n_i^*} |F_k^{(i)}(d_j) - \tilde{F}_k^{(i)}(d_j)|.$$

Denote $\Delta_1 = \frac{1}{m_k^{(i)}}$, $\Delta_2 = \frac{1}{q_k^{(i)}}$. According to the definition of CDF, $\mathcal{F}$ can be expressed as a function of $x_1, x_2 \in \mathbb{N}^+$:

$$\mathcal{F} = f(x_1, x_2) = x_1\Delta_1 - x_2\Delta_2.$$

We analyze $f(x_1, x_2)$ based on the following conditions:

(i). $n_k^{(i)} \geq q_k^{(i)}$ and $\tilde{\mathcal{J}}(\boldsymbol{s}_i) = \emptyset$:

In this case, $\Delta_1 \leq \Delta_2$, $\mathcal{J}(\boldsymbol{s}_i) = \tilde{O}(\boldsymbol{s}_i) \subset O(\boldsymbol{s}_i)$. We first prove $|f(x_1, x_2)|$ attains its maximum when x_2 reaches to its minimum or maximum.

Since $\mathcal{J}(\boldsymbol{s}_i) \subset O(\boldsymbol{s}_i)$, we have $x_1 \geq x_2$ naturally. On the other hand, we get

$$f(x_1, x_2) = \frac{x_1}{m_k^{(i)}} + \frac{x_2}{q_k^{(i)}} = x_1(\Delta_1 - \Delta_2) + \Delta_2(x_1 - x_2). \tag{5.30}$$

Here the first term is negative, and the second term is positive. We consider the following conditions:

(1a). $\frac{x_1}{x_2} \geq \frac{m_k^{(i)}}{q_k^{(i)}}$: we have $f(x_1, x_2) \geq 0$. From equation (5.30), smaller x_2 makes the positive term larger. Thus, $f(x_1, x_2)$ gets larger, and so does $|f(x_1, x_2)|$.

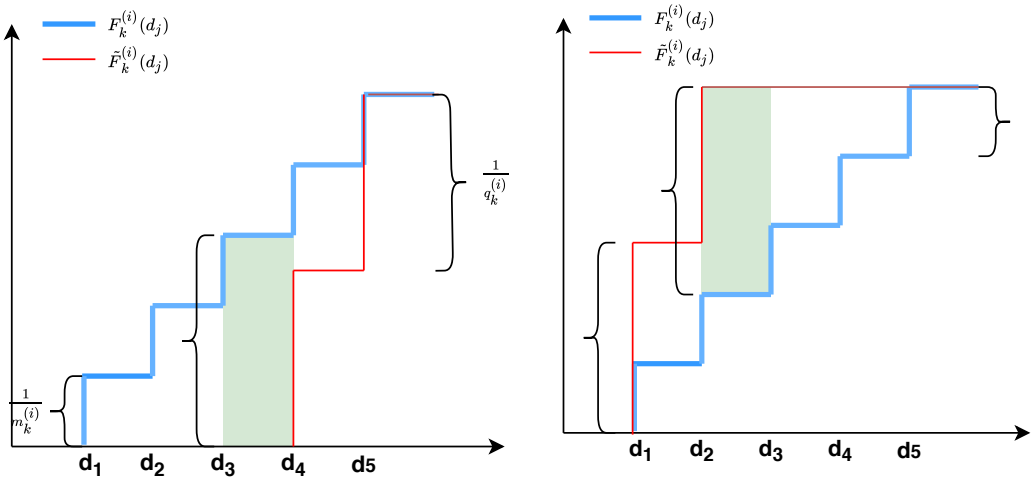
(1b). $\frac{x_1}{x_2} < \frac{m_k^{(i)}}{q}$: we have $f(x_1, x_2) < 0$. Larger x_2 makes the positive term smaller. Thus, $f(x_1, x_2)$ gets smaller, and $|f(x_1, x_2)|$ gets larger. Thus, the maximum value of $|f(x_1, x_2)|$ will be reached at either the minimum or the maximum of x_2. This naturally leads us to calculate the upper bound of $\mathcal{F}$ under the following two cases:

(1c). $x_2 = 0$: the maximum of x_1 would be obtained at $x_1 = m_k^{(i)} - q_k^{(i)}$. Figure 5.1a illustrate this scenario. The blue and red line stand for $F_k^{(i)}(d_j)$ and $\tilde{F}_k^{(i)}(d_j)$ respectively. $\mathcal{F}$ attains the maximum when all values in $\mathcal{J}(\boldsymbol{s}_i)$ comes from the large values in $O(\boldsymbol{s}_i)$ so that we can take the difference (the height of the green area) as right as possible. In this case, $\mathcal{F} \leq (m_k^{(i)} - q_k^{(i)})\Delta_1 = 1 - \frac{q_k^{(i)}}{m_k^{(i)}}$.

(1d). $x_2 = q_k^{(i)}$: the smallest x_1 is $x_1 = q_k^{(i)}$. Figure 5.1b shows the maximum difference under this scenario. $\mathcal{F}$ attains its maximum when all values in $\mathcal{J}(\boldsymbol{s}_i)$ comes from the small values inx $O(\boldsymbol{s}_i)$ so that we can take the difference (the height of the green area) as left as possible. In this case, $\mathcal{F} \leq q(\Delta_2 - \Delta_1) = 1 - \frac{q_k^{(i)}}{m_k^{(i)}}$.

The above analysis indicates that $\mathcal{F}$ will not exceed $1 - \frac{q_k^{(i)}}{m_k^{(i)}}$ in both cases.

(ii). $m_k^{(i)} \geq q_k^{(i)}$ and $\tilde{\mathcal{J}}(\boldsymbol{s}_i) \neq \emptyset$:



(a) Scenario (1c).

(b) Scenario (1d).

Figure 5.1: Upper Bound for Different Scenarios.

In this case, we can divide the problem into two sub-problems:

(2a). $d \leq max_{d_{ij}} O(s_i)$: all distance points come from $O(s_i)$. Our previous statement still holds: $|f(x_1, x_2)|$ attains its maximum at x_2's extreme value. The only difference with case (i) is x_2 cannot be $q_k^{(i)}$ any more because of the fact that $\tilde{\mathcal{J}}(s_i) \neq \emptyset$. There are $\tilde{q}_k^{(i)}$ points in $O(s_i)$ contained in $\mathcal{J}(s_i)$. When $x_2 = 0$, it is still possible to take $x_1 = m_k^{(i)} - \tilde{m}_k^{(i)}$ such that $\mathcal{F} = 1 - \frac{\tilde{m}_k^{(i)}}{m_k^{(i)}}$.

(2b). $d > max_{d_{ij}} O(s_i)$: the empirical distribution of $O(s_i)$ will not change and always equal to 1. Thus, the upper bound of $\mathcal{F}$ is obtained when x_1 is small. Namely, $x_2 = \tilde{n}_k^{(i)} \cdot \Delta_2 = \frac{\tilde{n}_k^{(i)}}{q_k^{(i)}}$. Figure 5.2a shows that the upper bound case would be $1 - \frac{\tilde{n}_k^{(i)}}{q_k^{(i)}}$.

The above two cases indicate that $\mathcal{F}$ would not exceed $1 - \frac{\tilde{m}_k^{(i)}}{q_k^{(i)}}$. Since $m_k^{(i)} > q_k^{(i)}$. Thus, condition (i) in Theorem 4.2.2 holds true.

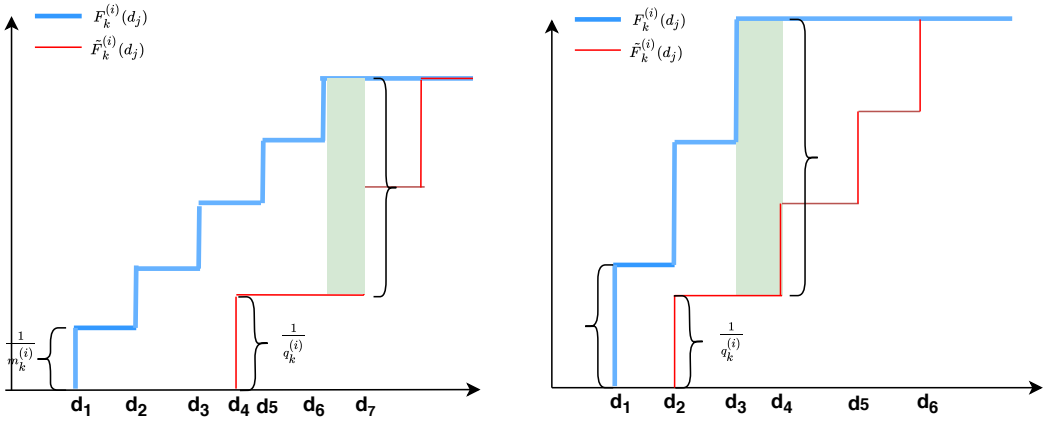
(iii). $m_k^{(i)} < q_k^{(i)}$:

It would be impossible that $\tilde{\mathcal{J}}(s_i) = \emptyset$ in this scenario. We can analyze $\mathcal{F}$ in two cases:

(3a). $d > max_{d_{ij}} O(s_i)$: it is similar with case (2b). $\mathcal{F}$ attains its maximum when x_2 is small, which is $1 - \frac{\tilde{m}_k^{(i)}}{q_k^{(i)}}$.

(3b). $d \leq max_{d_{ij}} O(s_i)$: since $\Delta_1 > \Delta_2$ and $x_1 \geq x_2$ in this case. The two terms in

equation (5.30) would both be positive. On the other hand, when x_1 increase 1, x_2 would also increase 1 at most. Thus, increasing x_1 will always make $f(x_1, x_2)$ larger. Thus, the maximum empirical difference is attained at x_1's maximum. The scenario is illustrated in Figure 5.2b. Thus, we get the maximum of $\mathcal{F}$ to be $1 - \frac{\tilde{n}_k^{(i)}}{q_k^{(i)}}$.



(a) Scenario (2b). (b) Scenario (3b).

Figure 5.2: Upper Bound for Different Scenarios.

Combining the two cases together, we have $\mathcal{F} = 1 - \frac{\tilde{n}_k^{(i)}}{q_k^{(i)}}$, which leads to the conclusion in Theorem 4.2.2 (ii). □

A.7 Proof of Theorem 4.3.1

Proof. Due to the IID assumption of the data and distributional symmetry of $R_{n+1,i}$ and $R_{i,n+1}$ $(j = 1, \ldots, n)$, we have

$$\mathbb{E}\{1_{\{\sum_{i\in\mathcal{B}_j} 1_{\{R_{ij}\geq R_{ji}\}}\geq \alpha n\}}\} = \mathbb{E}\{1_{\{\sum_{i=1}^{n} 1_{\{R_{n+1,i}\geq R_{i,n+1}\}}\geq \alpha n\}}\} \tag{5.31}$$

where $\mathcal{B}_j = \{i : s_i \in \mathcal{D}^*, i \neq j\}$. By definition of $V(s_{n+1})$

$$\begin{aligned}
&\mathbb{P}(s_{n+1} \in \{x : V(x) \geq \alpha\}) \\
=&\mathbb{E}1_{\{\sum_{i=1}^{n} 1_{\{R_{n+1,i} \geq R_{i,n+1}\} \geq \alpha n\}}} \\
=&\frac{1}{n}\sum_{j=1}^{n} \mathbb{E}1_{\{\sum_{i\in\mathcal{B}_j} 1_{\{R_{j,i}\geq R_{i,j}\}\geq \alpha n\}}} = \frac{1}{n}\mathbb{E}\{N_n\},
\end{aligned} \tag{5.32}$$

where $N_n = \sum_{j=1}^{n} \mathbb{E}1_{\{\sum_{i\in\mathcal{B}_j} 1_{\{R_{j,i}\geq R_{i,j}\geq \alpha n\}}} = \frac{1}{n}\mathbb{E}\{N_n\}$ is the size of the set $\mathcal{J} = \{j' | \sum_{i\in\mathcal{B}_{j'}} 1_{\{R_{j',i}\geq R_{i,j'}\}\geq \alpha n}\}$.

Consider for $\forall j \notin \mathcal{J}$, it satisfies

$$\begin{aligned}
\alpha n >& \sum_{i\in\mathcal{B}_j} 1_{\{R_{j,i}\geq R_{i,j}\}} = \sum_{i\in\mathcal{B}_j \bigcap \mathcal{J}} 1_{\{R_{j,i}\geq R_{i,j}\}} + \sum_{i\in\mathcal{B}_j\setminus\mathcal{J}} 1_{\{R_{j,i}\geq R_{i,j}\}} \\
=& \sum_{i\in\mathcal{B}_j\setminus\mathcal{J}} 1_{\{R_{j,i}\geq R_{i,j}\}}.
\end{aligned} \tag{5.33}$$

Summing over all $j \notin \mathcal{J}$, which has $n - N_n$ numbers, we have

$$\begin{aligned}
\alpha n(n - N_n) >& \sum_{j\notin\mathcal{J}} \sum_{i\in\mathcal{B}_j\setminus\mathcal{J}} 1_{\{R_{ji}\geq R_{ij}\}} = \sum_{j\notin\mathcal{J}, i\notin\mathcal{J}, i\neq j} 1_{\{R_{ji}\geq R_{ij}\}} \\
=& \frac{(n - N_n)(n - N_n - 1)}{2}.
\end{aligned} \tag{5.34}$$

Solving the above inequality, we get a lower bound on $N_n \geq (1 - 2\alpha)n$. By equation (5.32), it follows that $P(s_{n+1} \in \{x : V(x) \geq \alpha\}) = \frac{1}{n}\mathbb{E}\{N_n\} \geq 1 - 2\alpha$. □

REFERENCES

Abdulshahed, Ali M, Andrew P Longstaff, Simon Fletcher, and Alan Myers. 2015. "Thermal error modelling of machine tools based on ANFIS with fuzzy c-means clustering using a thermal imaging camera." *Applied Mathematical Modelling* 39 (7): 1837–1852.

Aggarwal, Charu C. 2017. "High-dimensional outlier detection: The subspace method." In *Outlier Analysis,* 149–184. Springer.

Aggarwal, Charu C, and S Yu Philip. 2005. "An effective and efficient algorithm for high-dimensional outlier detection." *The VLDB journal* 14 (2): 211–221.

Aggarwal, Charu C, and Saket Sathe. 2015. "Theoretical foundations and algorithms for outlier ensembles." *Acm Sigkdd Explorations Newsletter* 17 (1): 24–47.

Agrawal, Rakesh, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. 1998. "Automatic subspace clustering of high dimensional data for data mining applications." In *Proceedings of the 1998 ACM SIGMOD international conference on Management of data,* 94–105.

Akaike, Hirotogu. 1998. "Information theory and an extension of the maximum likelihood principle." In *Selected papers of hirotugu akaike,* 199–213. Springer.

Anderberg, MR. 1973. "Cluster analysis for researchers." *New York.*

Angiulli, Fabrizio, and Fabio Fassetti. 2007. "Detecting distance-based outliers in streams of data." In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management,* 811–820.

Angiulli, Fabrizio, and Clara Pizzuti. 2002. "Fast outlier detection in high dimensional spaces." In *European conference on principles of data mining and knowledge discovery,* 15–27. Springer.

Arning, Andreas, Rakesh Agrawal, and Prabhakar Raghavan. 1996. "A Linear Method for Deviation Detection in Large Databases." In *KDD,* 1141:972–981. 50.

Ball, Geoffrey H, and David J Hall. 1965. *ISODATA, a novel method of data analysis and pattern classification.* Technical report. Stanford research inst Menlo Park CA.

Barber, Rina Foygel, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani. 2019a. "Predictive inference with the jackknife+." *arXiv preprint arXiv:1905.02928.*

Barber, Rina Foygel, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani. 2019b. "The limits of distribution-free conditional predictive inference." *arXiv preprint arXiv:1903.04684.*

Bauschke, Heinz H, Patrick L Combettes, et al. 2011. *Convex analysis and monotone operator theory in Hilbert spaces.* Vol. 408. Springer.

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

Behera, HS, Abhishek Ghosh, and S Ku Mishra. 2012. "A new hybridized K-Means clustering based outlier detection technique for effective data mining." *International Journal of Advanced Research in Computer Science and Software Engineering:* 287–292.

Bezdek, James C. 1973. "Cluster validity with fuzzy sets."

Bhat, Harish S, and Nitesh Kumar. 2010. "On the derivation of the Bayesian Information Criterion." *School of Natural Sciences, University of California* 99.

Boyd, Stephen, Neal Parikh, and Eric Chu. 2011. *Distributed optimization and statistical learning via the alternating direction method of multipliers.* Now Publishers Inc.

Breunig, Markus M, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. "LOF: identifying density-based local outliers." In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data,* 93–104.

Cao, Feng, Martin Estert, Weining Qian, and Aoying Zhou. 2006. "Density-based clustering over an evolving data stream with noise." In *Proceedings of the 2006 SIAM international conference on data mining,* 328–339. SIAM.

Cellini, Roberto, and Tiziana Cuccia. 2013. "Museum and monument attendance and tourism flow: A time series analysis approach." *Applied Economics* 45 (24): 3473–3482.

Chavez, MJ. 2006. "Nuclear Waste Management Procedure NP 19-1: Software Requirements, Revision 11." *Section* 2 (1): 7–8.

Chen, Gary K, Eric C Chi, John Michael O Ranola, and Kenneth Lange. 2015a. "Convex clustering: An attractive alternative to hierarchical clustering." *PLoS computational biology* 11 (5): e1004228.

———. 2015b. "Convex clustering: An attractive alternative to hierarchical clustering." *PLoS Comput Biol* 11 (5): e1004228.

Chi, E. C., and K. Lange. 2013. "Splitting Methods for Convex Clustering." *ArXiv e-prints* (April). arXiv: `1304.0499 [stat.ML]`.

Chowdhury, AKM Rasheduzzaman, Md Elias Mollah, and Md Asikur Rahman. 2010. "An efficient method for subjectively choosing parameter ?k?automatically in VDBSCAN (Varied Density Based Spatial Clustering of Applications with Noise) algorithm." In *2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE),* 1:38–41. IEEE.

Combettes, Patrick L, and Jean-Christophe Pesquet. 2011. "Proximal splitting methods in signal processing." In *Fixed-point algorithms for inverse problems in science and engineering,* 185–212. Springer.

Cormack, Richard M. 1971. "A review of classification." *Journal of the Royal Statistical Society: Series A (General)* 134 (3): 321–353.

D'Andrade, Roy G. 1978. "U-statistic hierarchical clustering." *Psychometrika* 43 (1): 59–67.

Dimitrova, DS, VK Kaishev, and S Tan. 2017. "Computing the Kolmogorov-Smirnov distribution when the underlying cdf is purely discrete, mixed or continuous."

Ding, Chris, and Xiaofeng He. 2002. "Cluster merging and splitting in hierarchical clustering algorithms." In *2002 IEEE International Conference on Data Mining, 2002. Proceedings.* 139–146. IEEE.

Dutta, Jayanta K, Bonny Banerjee, and Chandan K Reddy. 2015. "RODS: Rarity based outlier detection in a sparse coding framework." *IEEE Transactions on Knowledge and Data Engineering* 28 (2): 483–495.

Ertöz, Levent, Michael Steinbach, and Vipin Kumar. 2003. "Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data." In *Proceedings of the 2003 SIAM international conference on data mining,* 47–58. SIAM.

Everitt, Brian S, Sabine Landau, Morven Leese, and Daniel Stahl. 2011. *Cluster analysis 5th ed.*

Fan, Hongqin, Osmar R Zaïane, Andrew Foss, and Junfeng Wu. 2006. "A nonparametric outlier detection for effectively discovering top-n outliers from engineering data." In *Pacific-Asia conference on knowledge discovery and data mining,* 557–566. Springer.

Friedman, Herman P, and Jerrold Rubin. 1967. "On some invariant criteria for grouping data." *Journal of the American Statistical Association* 62 (320): 1159–1178.

Frühwirth-Schnatter, Sylvia. 2006a. *Finite mixture and Markov switching models.* Springer Science & Business Media.

Frühwirth-Schnatter, Sylvia. 2006b. *Finite mixture and Markov switching models.* Springer Science & Business Media.

Gabay, Daniel, and Bertrand Mercier. 1975. *A dual algorithm for the solution of non linear variational problems via finite element approximation.* Institut de recherche d'informatique et d'automatique.

Ghoting, Amol, Srinivasan Parthasarathy, and Matthew Eric Otey. 2008. "Fast mining of distance-based outliers in high-dimensional datasets." *Data Mining and Knowledge Discovery* 16 (3): 349–364.

Glowinski, Roland, and A Marroco. 1975. "Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires." *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique* 9 (R2): 41–76.

Goil, Sanjay, Harsha Nagesh, and Alok Choudhary. 1999. *MAFIA: Efficient and Scalable Clustering for very large data sets: Technical Report No.* Technical report. CPDC–TR–9906–010© 1999 Center for Parallel and distributed Computing.

GROVE, WILLIAM M. 1984. "Cluster Analysis for Social Scientists." *American Journal of Psychiatry* 141 (10): 1297–a.

Guha, Sudipto, Rajeev Rastogi, and Kyuseok Shim. 1998. "CURE: an efficient clustering algorithm for large databases." *ACM Sigmod record* 27 (2): 73–84.

Guha, Sudipto, Adam Meyerson, Nina Mishra, Rajeev Motwani, and Liadan O'Callaghan. 2003. "Clustering data streams: Theory and practice." *IEEE transactions on knowledge and data engineering* 15 (3): 515–528.

Hartigan, John A. 1975. *Clustering algorithms.* John Wiley & Sons, Inc.

Hautamaki, Ville, Ismo Karkkainen, and Pasi Franti. 2004. "Outlier detection using k-nearest neighbour graph." In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* 3:430–433. IEEE.

Hennig, Christian. 2015. "What are the true clusters?" *Pattern Recognition Letters* 64:53–62.

Herdin, Markus, Nicolai Czink, Hüseyin Ozcelik, and Ernst Bonek. 2005. "Correlation matrix distance, a meaningful measure for evaluation of non-stationary MIMO channels." In *2005 IEEE 61st Vehicular Technology Conference,* 1:136–140. IEEE.

Hestenes, Magnus R. 1969. "Multiplier and gradient methods." *Journal of optimization theory and applications* 4 (5): 303–320.

Hocking, Toby Dylan, Armand Joulin, Francis Bach, and Jean-Philippe Vert. 2011. "Clusterpath an algorithm for clustering using convex fusion penalties." In *28th international conference on machine learning,* 1.

Ingrassia, Salvatore, Simona C Minotti, and Antonio Punzo. 2014. "Model-based clustering via linear cluster-weighted models." *Computational Statistics & Data Analysis* 71:159–182.

Ji, Zexuan, Jinyao Liu, Guo Cao, Quansen Sun, and Qiang Chen. 2014. "Robust spatially constrained fuzzy c-means algorithm for brain MR image segmentation." *Pattern recognition* 47 (7): 2454–2466.

Johnson, Stephen C. 1967. "Hierarchical clustering schemes." *Psychometrika* 32 (3): 241–254.

Johnson, Theodore, Ivy Kwok, and Raymond T Ng. 1998. "Fast Computation of 2-Dimensional Depth Contours." In *KDD,* 224–228. Citeseer.

Kannan, SR, S Ramathilagam, R Devi, and Evor Hines. 2012. "Strong fuzzy c-means in medical image data analysis." *Journal of Systems and Software* 85 (11): 2425–2438.

Karypis, George, Eui-Hong Han, and Vipin Kumar. 1999. "Chameleon: Hierarchical clustering using dynamic modeling." *Computer* 32 (8): 68–75.

Knorr, Edwin M, and Raymond T Ng. 1997. "A unified approach for mining outliers." In *Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research,* 11. IBM Press.

———. 1999. "Finding intensional knowledge of distance-based outliers." In *VLDB,* 99:211–222.

Knox, Edwin M, and Raymond T Ng. 1998. "Algorithms for mining distancebased outliers in large datasets." In *Proceedings of the international conference on very large data bases,* 392–403. Citeseer.

Kriegel, Hans-Peter, Peer Kroger, Erich Schubert, and Arthur Zimek. 2011. "Interpreting and unifying outlier scores." In *Proceedings of the 2011 SIAM International Conference on Data Mining,* 13–24. SIAM.

Kriegel, Hans-Peter, Peer Kröger, Erich Schubert, and Arthur Zimek. 2009a. "LoOP: local outlier probabilities." In *Proceedings of the 18th ACM conference on Information and knowledge management,* 1649–1652.

Kriegel, Hans-Peter, Peer Kröger, Erich Schubert, and Arthur Zimek. 2009b. "Outlier detection in axis-parallel subspaces of high dimensional data." In *Pacific-Asia Conference on Knowledge Discovery and Data Mining,* 831–838. Springer.

Kullback, Solomon, and Richard A Leibler. 1951. "On information and sufficiency." *The annals of mathematical statistics* 22 (1): 79–86.

Kutbay, Uğurhan, and Fırat Hardalaç. 2017. "Development of a multiprobe electrical resistivity tomography prototype system and robust underground clustering." *Expert Systems* 34 (3): e12206.

Kutbay, Uğurhan, Ali Berkan Ural, and Fırat Hardalaç. 2015. "Underground electrical profile clustering using K-MEANS algorithm." In *2015 23nd Signal Processing and Communications Applications Conference (SIU),* 561–564. IEEE.

Lance, Godfrey N, and William Thomas Williams. 1967. "A general theory of classificatory sorting strategies: 1. Hierarchical systems." *The computer journal* 9 (4): 373–380.

Lazarevic, Aleksandar, and Vipin Kumar. 2005. "Feature bagging for outlier detection." In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining,* 157–166.

Lei, Jing, Max GSell, Alessandro Rinaldo, Ryan J Tibshirani, and Larry Wasserman. 2018. "Distribution-free predictive inference for regression." *Journal of the American Statistical Association* 113 (523): 1094–1111.

Lindsten, Fredrik, Henrik Ohlsson, and Lennart Ljung. 2011. *Just relax and come clustering!: A convexification of k-means clustering.* Linköping University Electronic Press.

MacQueen, James, et al. 1967. "Some methods for classification and analysis of multivariate observations." In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability,* 1:281–297. 14. Oakland, CA, USA.

Massey Jr, Frank J. 1951. "The Kolmogorov-Smirnov test for goodness of fit." *Journal of the American statistical Association* 46 (253): 68–78.

McLachlan, Geoffrey J, and David Peel. 2004. *Finite mixture models.* John Wiley & Sons.

McNicholas, Paul D. 2016. "Model-based clustering." *Journal of Classification* 33 (3): 331–373.

Milenova, Boriana L, and Marcos M Campos. 2002. "O-cluster: Scalable clustering of large high dimensional data sets." In *2002 IEEE International Conference on Data Mining, 2002. Proceedings.* 290–297. IEEE.

Milligan, Glenn W. 1979. “Ultrametric hierarchical clustering algorithms.” *Psychometrika* 44 (3): 343–346.

minFunc: unconstrained differentiable multivariate optimization in Matlab.

Moro, Sérgio, Paulo Cortez, and Paulo Rita. 2014. “A data-driven approach to predict the success of bank telemarketing.” *Decision Support Systems* 62:22–31.

Müller, Emmanuel, Matthias Schiffer, and Thomas Seidl. 2010. “Adaptive outlierness for subspace outlier ranking.” In *Proceedings of the 19th ACM international conference on Information and knowledge management,* 1629–1632.

———. 2011. “Statistical selection of relevant subspace projections for outlier ranking.” In *2011 IEEE 27th international conference on data engineering,* 434–445. IEEE.

Muller, Emmanuel, Ira Assent, Uwe Steinhausen, and Thomas Seidl. 2008. “OutRank: ranking outliers in high dimensional data.” In *2008 IEEE 24th international conference on data engineering workshop,* 600–603. IEEE.

Parikh, Neal, Stephen Boyd, et al. 2014. “Proximal algorithms.” *Foundations and Trends® in Optimization* 1 (3): 127–239.

Parimala, M, Daphne Lopez, and NC Senthilkumar. 2011. “A survey on density based clustering algorithms for mining large spatial databases.” *International Journal of Advanced Science and Technology* 31 (1): 59–66.

Qiu, Cunyong, Jian Xiao, Long Yu, Lu Han, and Muhammad Naveed Iqbal. 2013. “A modified interval type-2 fuzzy C-means algorithm with application in MR image segmentation.” *Pattern Recognition Letters* 34 (12): 1329–1338.

Ramaswamy, Sridhar, Rajeev Rastogi, and Kyuseok Shim. 2000. “Efficient algorithms for mining outliers from large data sets.” In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data,* 427–438.

Rand, William M. 1971a. “Objective criteria for the evaluation of clustering methods.” *Journal of the American Statistical association* 66 (336): 846–850.

———. 1971b. “Objective criteria for the evaluation of clustering methods.” *Journal of the American Statistical association* 66 (336): 846–850.

Rockafellar, R Tyrell. 1973. “The multiplier method of Hestenes and Powell applied to convex programming.” *Journal of Optimization Theory and applications* 12 (6): 555–562.

Ruts, Ida, and Peter J Rousseeuw. 1996. "Computing depth contours of bivariate point clouds." *Computational Statistics & Data Analysis* 23 (1): 153–168.

Sakoe, Hiroaki, and Seibi Chiba. 1978. "Dynamic programming algorithm optimization for spoken word recognition." *IEEE transactions on acoustics, speech, and signal processing* 26 (1): 43–49.

Shah, Glory H, CK Bhensdadia, and Amit P Ganatra. 2012. "An empirical evaluation of density-based clustering techniques." *International Journal of Soft Computing and Engineering (IJSCE) ISSN* 22312307:216–223.

She, Tarafdar, and Zhang. 2020. "Supervised Multivariate Learning with Simultaneous Feature Auto-grouping and Dimension Reduction."

She, Yiyuan. 2012. "An iterative algorithm for fitting nonconvex penalized generalized linear models with grouped predictors." *Computational Statistics & Data Analysis* 56 (10): 2976–2990.

She, Yiyuan, et al. 2010. "Sparse regression with exact clustering." *Electronic Journal of Statistics* 4:1055–1096.

———. 2009. "Thresholding-based iterative selection procedures for model selection and shrinkage." *Electronic Journal of statistics* 3:384–415.

Sheikholeslami, Gholamhosein, Surojit Chatterjee, and Aidong Zhang. 1998. "Wavecluster: A multi-resolution clustering approach for very large spatial databases." In *VLDB,* 98:428–439.

Sneath, PHA. 1977. "A method for testing the distinctness of clusters: a test of the disjunction of two clusters in euclidean space as measured by their overlap." *Journal of the International Association for Mathematical Geology* 9 (2): 123–143.

Späth, Helmuth. 1980. "Cluster analysis algorithms for data reduction and classification of objects."

Stahl, Daniel, and Hannah Sallis. 2012. "Model-based cluster analysis." *Wiley Interdisciplinary Reviews: Computational Statistics* 4 (4): 341–358.

Strehl, Alexander, and Joydeep Ghosh. 2002. "Cluster ensembles—a knowledge reuse framework for combining multiple partitions." *Journal of machine learning research* 3 (Dec): 583–617.

Tang, Jian, Zhixiang Chen, Ada Wai-Chee Fu, and David W Cheung. 2002. "Enhancing effectiveness of outlier detections for low density patterns." In *Pacific-Asia Conference on Knowledge Discovery and Data Mining,* 535–548. Springer.

Tibshirani, Robert. 2011. "Regression shrinkage and selection via the lasso: a retrospective." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 73 (3): 273–282.

Tong, Howell, and Keng S Lim. 2009. "Threshold autoregression, limit cycles and cyclical data." In *Exploration Of A Nonlinear World: An Appreciation of Howell Tong's Contributions to Statistics,* 9–56. World Scientific.

Tseng, Paul. 1991. "Applications of a splitting algorithm to decomposition in convex programming and variational inequalities." *SIAM Journal on Control and Optimization* 29 (1): 119–138.

Tsybakov, Alexandre B. 2008. *Introduction to nonparametric estimation.* Springer Science & Business Media.

Tukey, John W. 1977. *Exploratory data analysis.* Vol. 2. Reading, Mass.

Van Lint, J, and Gerard Van der Geer. 2012. *Introduction to coding theory and algebraic geometry.* Vol. 12. Birkhäuser.

Vovk, Vladimir, Alex Gammerman, and Glenn Shafer. 2005. *Algorithmic learning in a random world.* Springer Science & Business Media.

Wang, Wei, Jiong Yang, and Richard Muntz. n.d. "STING: A Statistical Grid Approach to Spatial Data Mining: Department of Computer Science." *University of California, Los Angels.*

Ward Jr, Joe H. 1963. "Hierarchical grouping to optimize an objective function." *Journal of the American statistical association* 58 (301): 236–244.

Wolfe, John H. 1970. "Pattern clustering by multivariate mixture analysis." *Multivariate Behavioral Research* 5 (3): 329–350.

Wright, Stephen, and Jorge Nocedal. 1999. "Numerical optimization." *Springer Science* 35 (67-68): 7.

Wu, Senlin, and Rong Chen. 2007. "Threshold variable determination and threshold variable driven switching autoregressive models." *Statistica Sinica* 17 (1): 241–S38.

Xie, Min-ge, and Zheshi Zheng. 2020a. "Discussion of Professor Bradley Efrons Article on Prediction, Estimation, and Attribution." *Journal of the American Statistical Association* 115 (530): 667–671.

Xie, Min-ge, and Zheshi Zheng. 2020b. "Homeostasis phenomenon in predictive inference when using a wrong learning model: a tale of random split of data into training and test sets." *arXiv preprint arXiv:2003.08989.*

Zadeh, Lotfi A. 1965. "Fuzzy sets." *Information and control* 8 (3): 338–353.

Zhang, Jifu, Yiyong Jiang, Kai H Chang, Sulan Zhang, Jianghui Cai, and Lihua Hu. 2009. "A concept lattice based outlier mining method in low-dimensional subspaces." *Pattern Recognition Letters* 30 (15): 1434–1439.

Zhang, Jifu, Xiaolong Yu, Yonghong Li, Sulan Zhang, Yaling Xun, and Xiao Qin. 2016. "A relevant subspace based contextual outlier mining algorithm." *Knowledge-Based Systems* 99:1–9.

Zhang, Jifu, Sulan Zhang, Kai H Chang, and Xiao Qin. 2014. "An outlier mining algorithm based on constrained concept lattice." *International Journal of Systems Science* 45 (5): 1170–1179.

Zhang, Tian, Raghu Ramakrishnan, and Miron Livny. 1996. "BIRCH: an efficient data clustering method for very large databases." *ACM sigmod record* 25 (2): 103–114.

Zimek, Arthur, Ricardo JGB Campello, and Jörg Sander. 2014. "Ensembles for unsupervised outlier detection: challenges and research questions a position paper." *Acm Sigkdd Explorations Newsletter* 15 (1): 11–22.

Zimek, Arthur, Erich Schubert, and Hans-Peter Kriegel. 2012. "A survey on unsupervised outlier detection in high-dimensional numerical data." *Statistical Analysis and Data Mining: The ASA Data Science Journal* 5 (5): 363–387.

Zimek, Arthur, Matthew Gaudet, Ricardo JGB Campello, and Jörg Sander. 2013. "Subsampling for efficient and effective unsupervised outlier detection ensembles." In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining,* 428–436.