Title
On the Adversarial Robustness of Machine Learning Algorithms

Author
Li, Fuwei

Publication Date
2021

Peer reviewed|Thesis/dissertation

On the Adversarial Robustness of Machine Learning Algorithms

By

FUWEI LI
DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Electrical and Computer Engineering

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

_____

Lifeng Lai, Chair

_____

Shuguang Cui

_____

Bernard C. Levy

Committee in Charge

2021

Abstract

Machine learning has been ubiquitously used in our daily lives. On the one hand, the success of machine learning depends on the availability of a large amount of data. On the other hand, the diverse data sources make a machine learning system harder to get very high quality data. What makes it worse is that there might be a malicious adversary who can deliberately modify the data or add poisoning data to corrupt the learning system. This imposes a great threat to the applications that are safety and security critical, for example, drug discovery, medical image analysis, and self-driving cars. Hence, it is necessary and urgent to investigate the behavior of machine learning under adversarial attacks. In this dissertation, we examine the adversarial robustness of three commonly used machine learning algorithms: linear regression, LASSO based feature selection, and principal component analysis (PCA).

In the first part, we study the adversarial robustness of linear regression. We assume there is an adversary in the linear regression system. The adversary tries to suppress or promote one of the regression coefficients. To obtain this goal, the adversary adds poisoning data samples or directly modifies the feature matrix of the original data. In the first scenario that the adversary intends to manipulate one of the regression coefficients by adding one carefully designed poisoning data, we derive the optimal form of the poisoning data. We also introduce a semidefinite relaxation method to design the poisoning data when the adversary tries to modify one of the regression coefficients while minimizing the changes of other regression coefficients. Finally, we propose an alternating optimization method to design the rank-one modification of the feature matrix.

In the second part, we extend the linear regression to LASSO based feature selection and study the best strategy to modify the feature matrix or response values to mislead the learning system to select the wrong features. We formulate this problem as a bi-level

optimization problem. As the ℓ_1 regularizer is not continuously differentiable, we use a smooth approximation of the ℓ_1 norm function and employ the interior point method to solve the LASSO problem and find the gradient information. Finally, we utilize the projected gradient descent method to design the modification strategy.

In the last part, we consider the adversarial robustness of the subspace learning problem. We examine the optimal modification strategy under the energy constraints to delude the PCA based subspace learning algorithm. Firstly, we derive the optimal rank-one attack strategy to modify the original data in order to maximize the subspace distance between the original one and the one after modification. Further, we do not constrict the rank of the modification and find the optimal modification strategy.

Acknowledgement

This dissertation and the work invested into it would not have been possible without the support and nurturing of many people.

First of all, I would like to express my deepest appreciation to my advisors: Prof. Lifeng Lai and Prof. Shuguang Cui. I am fortunate enough to have received help and guidance from two professors. I am extremely grateful to Prof. Lai. His insightful suggestion points me to the correct research direction. He always encourages me and gives me enough freedom to do the research that I am interested in. The weekly meeting with Prof. Lai is my most enjoyable time during my Ph.D. study. At each meeting, he constantly gives me inspiration for the research problems. I would also like to extend my sincere thanks to Prof. Cui. He gave me the opportunity to study at UC, Davis. He gives me sufficient guidance in academic research and teaches me how to improve my communication skills. This really helps me a lot. Without their encouragement and continuous guidance, I could not have finished this dissertation.

I am also grateful to my dissertation and qualification committee members, Prof. Bernard C. Levy, Prof. Xiaodong Li and Prof. Jinyi Qi. They gave me valuable advice on the choice of my research topics and insightful suggestions for the research problems from different perspectives. I also wish to thank Prof. Jun Fang, who brought me into the academic world during my graduate study.

I am deeply indebted to my family. None of my accomplishments would have been possible without the everlasting love and encouragement of my family. They have done all they could do to support me. My wife sacrifices herself and accompanies me at UC, Davis for more than four years. The birth of my beloved daughter brings my wife and me a lot of courage, joy, and challenge.

I would also like to thank all my friends and colleagues at UC, Davis, especially Songyang Zhang, Han Zhang, Hang Li, Qilian Yu, Chen Qiu, Zhi Wang, Yue Xu, Man Chu, Xiaochuan

Ma, Minhui Huang, Puning Zhao, Yulu Jin, Xinyang Cao, Xinyi Ni, Ying Li. I would also like to acknowledge the assistance of all my friends and teachers in Davis who made my five years here so wonderful.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we will first give an introduction to adversarial machine learning in Chapter 1.1. Then, we will introduce our motivation, the related works, and our contribution to linear regression, LASSO based feature selection and PCA based subspace learning in Chapters 1.2, 1.3 and 1.4, respectively.

1.1 Adversarial Machine Learning

Machine learning is being used in various applications. Most of the existing machine learning systems make the basic assumption that the data are from normal users and are generated independently from the same distribution. Even though there are algorithms designed to deal with small dense noises and large sparse outliers, few consider the adversarial noises. These noises are intentionally created by an adversary who has some knowledge of the machine learning system and the data. Then, the adversary will deliberately add some carefully designed noises or directly modify the data set in order to corrupt the learning system or mislead the learning system to make a wrong decision. This attack is especially dangerous for some security and safety critical applications such as medical image analysis [1] and autonomous driving [2].

Depending on the goal of the adversary, the adversarial attacks can be divided into

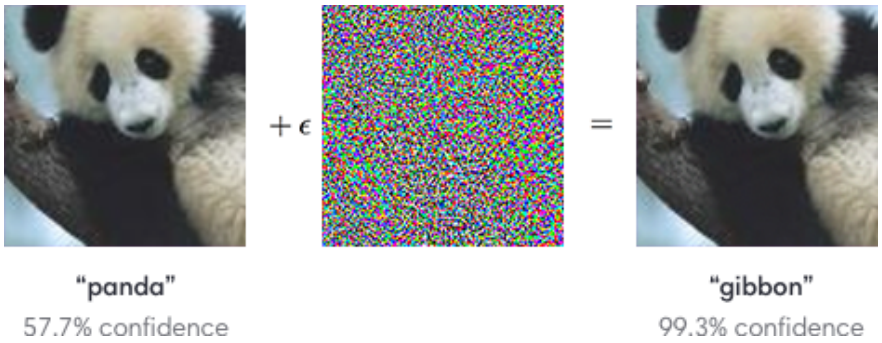


Figure 1.1: Demonstration of adversarial attack.

three categories: evasion, poisoning, model stealing. In the evasion attack, an adversary adds some imperceptible noises onto the original data and makes the learning system give a wrong prediction [3–5]. Fig. 1.1 demonstrates a typical evasion attack [3]. The original picture is a panda. The adversary adds some carefully designed noises onto it. Although it looks the same as the original panda, the classifier will miscategorize it as a gibbon. In the poisoning attack, the adversary attacks the learning systems by contaminating the training data. When the learning system train the model using the poisoned data, the model is then corrupted [6–11]. The adversary can also do model stealing by repeatedly sending requests to the server and then reconstruct the learning system or original training data. Model stealing also imposes great thread to the learning system that is sensitive and confidential [12, 13].

Depending on the adversary's knowledge about the data samples, the learning algorithm, and the defense strategy of the learning system, the adversary can carry out white-box, grey-box, and black-box attacks. In the white-box attack, the adversary has the full knowledge of the machine learning system and has the ability to observe the whole data points. After seeing the data points, the adversary can add some carefully designed poisoning data points or directly modify the data points so as to corrupt the learning system or leave a backdoor in this system [14]. If the adversary knows nothing about the data samples, learning algorithms, and defense strategies, the adversary can also carry out black-box attacks, where it gains information of the system by repeatedly sending queries to the system [15]. If the adversary only has partial knowledge of the data samples, learning algorithms, and defense strategies,

the adversary can perform grey-box attacks, in which it uses surrogate data samples or classifiers to mimic the original ones [16].

In this dissertation, we will focus on the white-box poisoning attack. Currently, most of the existing works concentrate on the deep learning based machine learning systems and propose some effective attack strategies upon that. However, due to the complexity of the deep learning system, we can only observe its effectiveness through their numerical demonstrations. We do not know whether their attacks are optimal. Besides, there is no theoretical performance guarantee for most of the attacks against the deep learning systems. Because of the lack of theory of deep learning, it is better to start from traditional machine learning algorithms and gain intuitions from their behavior under adversarial attacks. Hence, in this dissertation, we will study the adversarial robustness of three commonly used machine learning algorithms, i.e., linear regression, LASSO, and PCA.

1.2 Adversarial Attack Against Linear Regression

Linear regression plays a fundamental role in machine learning and is used in a wide spectrum of applications [17–21]. In linear regression, one assumes that there is a simple linear relationship between the explanatory variables and the response variable. The goal of linear regression is to find out the regression coefficients through the methods of ordinary least square (OLS):

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}} : \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2, \tag{1.1}$$

where $\mathbf{y} = [y_1, y_2, \ldots, y_n]^\top$ is the response values, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^\top \in \mathbb{R}^{n \times m}$ is the feature matrix, $\boldsymbol{\beta}$ is the regression coefficient, m is the number of explanatory variables, n is the number of data points, and $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ is the original data points. Having the regression coefficients learned from the data points, one can predict the response values given the values of the explanatory variables. The regression coefficients also help us explain the variation in

the response variable that can be attributed to the variation in the explanatory variables. They can quantify the strength of the relationship between certain explanatory variables and the response variable. A large magnitude of the regression coefficient usually indicates a strong relationship, while a small valued regression coefficient means a weak relationship. This is especially true when linear regression is accomplished by the parameter regularized method such as ridge regression and LASSO. In addition, the sign of the regression coefficients indicates whether the value of the response variable increases or decreases when the value of an explanatory variable changes, which is very important in biologic science [22], financial analysis [23], and environmental science [24].

Since the regression coefficient is very important, our work is to investigate the adversarial robustness of linear regression. In the considered linear regression system, there exists an adversary who can observe the whole dataset and then inject carefully designed poisoning data points or directly modify the original dataset in order to manipulate the regression coefficients. The manipulated regression coefficients can later be used by the adversary as a backdoor of this learning system or mislead our interpretation of the linear regression model. For example, by changing the magnitude of a regression coefficient to be small, it makes us believe that its corresponding explanatory variable is irrelevant. Similarly, the adversary can change the magnitude of a regression coefficient to a larger value to increase its importance. Furthermore, changing the sign of a regression coefficient can also lead us to misinterpret the correlation between its explanatory variable and the response variable.

We have several contributions to the adversarial attacks against linear regression in this dissertation. Depending on the objective of the adversary and the way the adversary changes the regression coefficients, we have different problem formulations. We first consider a scenario where the adversary tries to manipulate one specific regression coefficient by adding one carefully designed poisoning data point that has a limited energy budget to the dataset. We show that finding the optimal attack data point is equivalent to solve an optimization problem where the objective function is a ratio of two quadratic functions with a quadratic

inequality constraint. Even though this type of problem is non-convex in general, our particular problem has a hidden convex structure. With the help of this convex structure, we further convert the optimization problem into a quadratic constrained quadratic program (QCQP). Since strong duality exists in this problem [25], we manage to identify its closed-form optimal solutions from its Karush-Kuhn-Tucker (KKT) conditions.

We next consider a more sophisticated objective where the attacker aims to change one particular regression coefficient while making others be changed as small as possible. We show that the problem of finding the optimal attack data point is equivalent to solving an optimization problem where the objective function is a ratio of two fourth order multivariate polynomials with a quadratic inequality constraint. This optimization problem is much more complicated than the optimization above. We introduce a semidefinite relaxation method to solve this problem. The numerical examples show that we can find the globally optimal solutions with a very low relaxation order. Hence, the complexity of this method is low in practical problems.

Finally, we consider a more powerful adversary who can directly modify the feature matrix. Particularly, we consider a rank-one modification attack [26], where the attacker carefully designs a rank-one matrix and adds it to the existing data matrix. A rank-one modification attack is general enough to capture most of the common modifications, such as modifying one feature, deleting or adding one data point, changing one entry of the data matrix, etc. Hence, studying the rank-one modification provides us universal bounds on these kinds of attacks. By leveraging the rank-one structure, we develop an alternating optimization method to find the optimal modification matrix. We also prove that the solution obtained by the proposed optimization method is one of the critical points of the optimization problem.

Our study is related to several recent works on adversarial machine learning. For example, Pimentel-Alarcón et al. studied how to add one adversarial data point in order to maximize the error of the subspace estimated by principal component [27] and Li et al. derived a closed-

form optimal modification to the original dataset in order to maximize the subspace distance between the original one the one after modification [26]. These two works focused on the robustness of subspace learning algorithms that are based on PCA. PCA is an unsupervised learning method. By contrast, we study the robustness of linear regression, which is a supervised learning method. Alfeld et al. studied how to manipulate the training data so as to increase the validation or test error for the linear regression task [8, 9] and Biggio et al. used a gradient based algorithm to design one poisoning data point with the aim of worsening the testing error in a support vector machine (SVM) learning system and they also proposed a heuristic approach to flip parts of the training labels in order to achieve a similar goal [6,28]. These works aimed to deteriorate the performance of the machine learning system on a specific data set. However, we concentrate on the explanation of the linear regression model. By manipulating the regression coefficient, we can mislead the interpretation of the dependency between the features and response value. Furthermore, a series of works focused on the adversarial robustness of deep learning networks. Kurakin et al. proposed a gradient based method to design adversarial noise [3, 4, 29]. By adding this noise on the test data, it makes the machine learning system make the wrong prediction. By contrast, we focus on adding or modifying training data samples to maneuver the regression coefficient. Biggio et al. corrupted the deep learning system by inserting delicately designed poisoning data samples into the training data [11, 14, 30]. Due to the complexity of deep neural networks, it is hard to know whether the designed poisoning data samples are optimal. Nevertheless, our method is proven to be optimal with respect to certain specific goals discussed.

In addition, there are recent work that focus on the adversarial robustness of machine learning in various other applications. For example, Kwon et al. proposed a gradient based method to generate adversarial audio examples [31], Li et al. presented an ensemble method to enhance the robustness of the malware detection system against adversarial attacks [32], and Flowers et al. demonstrated the vulnerability of communication systems against adversarial noises [33]. These works are limited to their specific applications. Instead, we target

maneuvering the interpretation of a general linear regression model by adding poisoning data points or modifying the original data.

The most relevant work to ours is [34], where the authors develop a bi-level optimization framework to design the attack matrix. [34] used the projected gradient descent method to solve the bi-level optimization problem. However, a general bi-level problem is known to be NP hard and solving it depends on the convexity of the lower level problem. In addition, the convergence of projected gradient descent for a non-convex problem is not clear. Compared with [34], we obtain the globally optimal solution to the case for adding one poisoning data point, and we also prove that the proposed alternating optimization method converges to one of the critical points for the case where the attacker can perform a rank-one modification attack. Furthermore, for the projected gradient descent method, different datasets need different parameters, which means we must do parameter tuning before applying this algorithm. By contrast, we provide a closed-form solution to the case for adding one poisoning data point to attack one of the regression coefficients, and the designed alternating optimization method for the case of rank-one attack does not need parameter tuning. Furthermore, compared with the projected gradient descent method, our alternating optimization method provides smaller objective values, faster convergence rate, and more stable behavior.

The study of adversarial robustness of linear regression problem in our dissertation is based on our published and submitted papers [35, 36].

1.3 Adversarial Attack Against LASSO Based Feature Selection

Feature selection is one of the most important preprocessing steps in the vast majority of machine learning and signal processing problems [37–39]. By performing feature selection, we can discard irrelevant and redundant features while keeping the most informative features.

With the features of a smaller dimension, we can overcome the curse of dimensionality, better interpret our model, and speed up training and testing processes. Among a variety of feature selection methods, LASSO is one of the most widely used [40, 41]. LASSO can perform feature selection and regression simultaneously by solving the following ℓ_1 norm regularized least square problem:

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1, \tag{1.2}$$

where $\mathbf{y}$ and $\mathbf{X}$ are the response values and feature matrix respectively defined similarly to that in (1.1). Due to the sparse promotion ℓ_1 norm regularizer, most of the regression coefficients obtained by (1.2) will be zeros. The zero-valued coefficients correspondes to the features that are not chosen, while the non-zero valued coefficients indicate the selected features. Owing to its simplicity and efficiency, LASSO is widely applied to bio-science [42], financial analysis [43], image processing [44], etc. Furthermore, by exploring the additional structures of the regression coefficients, various extensions such as group LASSO [45, 46] and sparse group LASSO [47, 48] are proposed in the literature.

Since feature selection serves as the first stage of many of the machine learning algorithms, it is necessary and urgent to investigate its adversarial robustness. Though some existing works examined the robustness of feature selection against dense noise and outliers [49, 50], its behavior under the adversary attacks is unknown. By analyzing the attack strategy of the adversary, our goal is to provide a better understanding of the sensitivity of feature selection methods against this kind of attack.

In the considered feature selection model, we assume that there is an adversary who has the full knowledge of the model and can observe the whole dataset. After inspecting the dataset, it will carefully modify the response values or the feature matrix so as to manipulate the regression coefficients. By modifying the regression coefficients, it will maneuver the selected features. It can select the features which will not be selected originally by enlarging

the magnitude of the corresponding regression coefficients. Also, it can make us wrongly discard important features by suppressing the magnitude of the corresponding regression coefficients. Moreover, it will try to make other regression coefficients unchanged so as to minimize the possibility of being detected by the feature selection system. In this paper, we intend to find the best modification strategy of the adversary with the energy constraints on the modification. By doing so, we can better understand how the response values and feature matrix influence the selected features and the robustness of the feature selection algorithm.

We formulate this problem as a bi-level optimization problem. The upper-level objective is to minimize the difference between the targeted regression coefficients and that learned from the modified dataset. The lower-level problem is just a LASSO based feature selection problem with the modified dataset. To solve this bi-level optimization problem, we first solve the lower-level problem. Since the LASSO problem is a convex optimization problem, it is equivalent to its first order optimality condition. By applying the implicit function theorem on the first order optimality condition, we may learn the relationship between the dataset and the regression coefficients if the first order condition is continuously differentiable around its optimum. However, the ℓ_1 norm is not continuous at point zero. This prevents us from directly employing the implicit function theorem on the KKT conditions. To resolve the issue, we reformulate the LASSO problem as a linear inequality constrained quadratic programming problem and use the interior-point method to solve it. By utilizing the first order optimality condition from the reformulated problem, we are able to find the gradients of our objective with respect to the response values and feature matrix. With the gradients information, we employ the projected gradient descent to solve this bi-level optimization problem. Similar methods can be applied to design the attack strategy based on the group LASSO and the sparse group LASSO.

Our work of adversarial attack against LASSO based feature selection is based on our published and submitted papers [51, 52].

1.4 Adversarial Attack Against Subspace Learning

Subspace learning has a wide range of applications, such as surveillance video analysis, recommendation systems, anomaly detection, etc[53–60]. Among a large variety of subspace learning algorithms, principal component analysis is one of the most widely used ones. We will assume PCA the subspace learning algorithm. PCA computes a small number of principal components, which are orthogonal to each other and represent the majority of the variability of the data samples, and treats the span of these principal components as the desired low-dimensional subspace. Furthermore, many works have proposed robust PCA that can mitigate the impact of certain percentages of outliers and small dense random noise[61–64].

In Chapter 4 of our dissertation, we investigate the adversarial robustness of subspace learning algorithms. Particularly, we examine the robustness of subspace learning algorithms against not only random noise or unintentional corrupted data as considered in existing works but also malicious data produced by powerful adversaries who can modify the whole data set. Our study is motivated by the fact that subspace learning and many other machine learning algorithms are increasingly being used in safety critical and security related applications, such as autonomous vehicle system [65], voice recognition [66], medical image processing [1], etc. In these applications, there might exist powerful adversaries who can modify the data with the goal of maneuvering the machine learning algorithms to make the wrong decision or leave a backdoor in the system [14]. To ensure the security and safety of these systems, it is crucial to understand the impact of these adversarial attacks on the performance of machine learning algorithms.

In our problem, given the original data matrix, we learn a low-dimensional subspace via PCA. However, there is an adversary who can observe the whole data matrix and then carefully design a modification matrix to change the original data. The goal of the adversary is to modify the original data so as to maximize the subspace distance between the subspace learned from the original data and that learned from the modified data. In our dissertation,

we use Asimov distance[67], defined as the largest principal angle between two subspaces, to measure the subspace distance. Asimov distance has a close relationship with the chordal 2-norm distance and the Finsler distance, which are used in the analysis of optimization on manifolds [68, 69]. It is also related to the gap distance, which is used in the control theory to describe the stability and robustness of a system [70–72]. Additionally, it is closely connected to the projection 2-norm that is widely used in various applications[67,73,74]. The projection 2-norm provides a way to measure the discrepancy of the projections of a vector on two distinct subspaces. It is useful in the robustness analysis of the principal component regression (PCR), as one is actually projecting the response value vector onto the selected feature subspace in PCR. We will provide an example to illustrate it in Chapter 4.5 using real data. As the Asimov distance depends on the modification matrix in a complex manner, to characterize the optimal attack strategy that maximizes the Asimov distance, we need to solve a complicated non-convex optimization problem.

Towards this goal, we first solve the optimization problem with an additional rank-one constraint on the modification matrix. We note that a rank-one modification is already powerful enough to capture many common modifications such as changing one data sample, inserting one adversarial data point, deleting one feature, etc. Furthermore, the techniques and insights obtained from this special case are useful for the general case without the rank-one constraint. In the rank-one attack case, we study two different scenarios depending on whether the dimension of the selected subspace is equal to the rank of the data matrix or not. Our study reveals that the optimal attack strategy depends on the energy budget and the singular values of the data matrix. Specifically, in the scenario where the dimension of the selected subspace is the same as the rank of the data matrix, we show that the optimal rank-one strategy depends solely on the energy budget and the smallest singular value of the data matrix. In the scenario where the dimension of the selected subspace is less than the rank of the original data matrix, the optimal strategy depends not only on the energy budget but also on the kth and $(k+1)$th singular values, where k is the dimension of the

selected subspace.

Relying on the insights gained from the rank-one case, we then extend our study to the more general case where no rank constraint is imposed. Compared with the case with the rank-one constraint, the attacker now has a higher degree of freedom to modify the data, which makes the characterization of the optimal attack strategy significantly more challenging. To solve this optimization problem, we first prove that, under the basis of the principal components of the original data matrix, the optimal attack matrix only has a few non-zero entries at particular locations. This result greatly reduces the complexity of our problem. With the help of this result, we then simplify our problem to an optimization problem with the objective function being a ratio of two quadratic functions. To solve this non-convex problem, we further convert our optimization problem to a feasibility problem and find the closed-form solution to this problem. Our result shows that the optimal strategy depends on the energy budget and the kth and $(k+1)$th singular values of the data matrix. Our analysis shows that, compared with the optimal rank-one strategy, this strategy leads to a larger subspace distance.

Our study is related to the recent works on adversarial machine learning. For example, Jagielski et al. study how to change the data to manipulate the result of the regression learning system [9]. Lai et al. investigate the optimal modification strategy to maximize the inference errors in a multivariate estimation system [75]. In an interesting related work [27], Pimentel-Alarcón et al. study how to design an adversarial data sample and add it to the data matrix in order to maximize the Asimov distance between the subspace estimated by PCA from the contaminated data matrix and that from the original data matrix. [27] focuses on the case where the original data matrix is low-rank and the dimension of the selected subspace is equal to the rank of the data matrix. By contrast, we consider a more powerful adversarial setting, where the data matrix is not constrained to being low-rank, the dimension of the selected subspace does not necessarily equal the rank of the data matrix, and the adversary can modify the whole data matrix instead of only adding one data sample.

The work of adversarial attack against subspace learning is based on our published papers [26, 76, 77].

Chapter 2

Optimal Feature Manipulation Attacks Against Linear Regression

2.1 Introduction

In this chapter, we study the optimal feature manipulation attacks against linear regression. In particular, we study how to design poisoning data points and modify the feature matrix in order to manipulate the regression coefficient. This chapter is organized as follows. In Chapter 2.2, we consider the scenario where the attacker adds one carefully designed poisoning data point to the dataset. In Chapter 2.3, we investigate the rank-one attack strategy. Numerical examples are provided in Chapter 2.4 to illustrate the results we obtained in this paper. Finally, we provide concluding remarks in Chapter 2.5.

2.2 Attacking with One Adversarial Data Point

In this section, we consider the scenario where the attacker can add one carefully crafted data point to the existing dataset. We will extend the analysis to the case with more sophisticated attacks in Chapter 2.3.

2.2.1 Problem Formulation

Consider a dataset with n data samples, $\{y_i, \mathbf{x}_i\}_{i=1}^n$, where y_i is the response variable, $\mathbf{x}_i \in \mathbb{R}^m$ is the feature vector, where each component of $\mathbf{x}_i$ represents an explanatory variable. In this section, we consider an adversarial setup in which the adversary first observes the whole dataset $\{\mathbf{y}, \mathbf{X}\}$, in which $\mathbf{y} := [y_1, y_2, \ldots, y_n]^\top$ and $\mathbf{X} := [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^\top$, and then carefully designs an adversarial data point, $\{y_0, \mathbf{x}_0\}$, and adds it into the existing data samples. After inserting this adversarial data point, we have the poisoned dataset $\{\hat{\mathbf{y}}, \hat{\mathbf{X}}\}$, where $\hat{\mathbf{y}} := [y_0, y_1, y_2, \ldots, y_n]^\top$, $\hat{\mathbf{X}} := [\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^\top$.

From the dataset, we intend to learn a linear regression model. From the poisoned dataset, the learned model is obtained by solving

$$\underset{\boldsymbol{\beta}}{\text{argmin}} : \|\hat{\mathbf{y}} - \hat{\mathbf{X}}\boldsymbol{\beta}\|^2, \tag{2.1}$$

where $\|\cdot\|$ denotes the ℓ_2 norm for a vector and the induced 2-norm for a matrix throughout this chapter. Let $\hat{\boldsymbol{\beta}}$ be the optimal solution to problem (2.1). The goal of the adversary is to minimize some objective function, $f(\hat{\boldsymbol{\beta}})$, by carefully designing the adversarial data point. The form of $f(\hat{\boldsymbol{\beta}})$ depends on the specific goal of the attacker. For example, the attacker can try to reduce the importance of feature i by setting $f(\hat{\boldsymbol{\beta}}) = |\hat{\beta}_i|$, in which $\hat{\beta}_i$ is the ith component of $\hat{\boldsymbol{\beta}}$. Or the attacker can try to increase the importance of feature i by setting $f(\hat{\boldsymbol{\beta}}) = -|\hat{\beta}_i|$. To make the problem meaningful, in this chapter, we impose the energy constraint on the adversarial data point. Since one data point contains a feature vector and a response value, we put ℓ_2 norm constraint on the concatenated vector $[\mathbf{x}_0^\top, y_0]^\top$. With the objective $f(\hat{\boldsymbol{\beta}})$ and the energy constraint of the adversary data point, our problem can be

formulated as

$$\begin{aligned} \min_{\|[\mathbf{x}_0^\top, y_0]\| \leq \eta} : \quad & f(\hat{\boldsymbol{\beta}}) \\ \text{s.t.} \quad & \hat{\boldsymbol{\beta}} = \operatorname*{argmin}_{\boldsymbol{\beta}} : \|\hat{\mathbf{y}} - \hat{\mathbf{X}}\boldsymbol{\beta}\|^2, \end{aligned} \tag{2.2}$$

where η is the energy budget. The objective function, $f(\hat{\boldsymbol{\beta}})$, depends on the poisoning data point, $\{\mathbf{x}_0, y_0\}$, not in a direct way, but through a lower level optimization problem. What makes this problem even harder is the complication of the objective function. Depending on the goal of the adversary, the objective can be in various forms. In the following two subsections, we will discuss two important objectives and their solutions, respectively. The methods and insights obtained from these two cases could then be extended to cases with other objectives.

2.2.2 Attacking One Regression Coefficient

In this subsection, the goal of the adversary is to design the adversarial data point $\{y_0, \mathbf{x}_0\}$ to decrease (or increase) the importance of a certain explanatory variable. If the goal is to decrease the importance of explanatory variable i, we can set $f(\hat{\boldsymbol{\beta}}) = |\hat{\beta}_i|$, and the optimization problem can be written as

$$\begin{aligned} \min_{\|[\mathbf{x}_0^\top, y_0]\|_2 \leq \eta} : \quad & |\hat{\beta}_i| \\ \text{s.t.} \quad & \hat{\boldsymbol{\beta}} = \operatorname*{argmin}_{\boldsymbol{\beta}} : \|\hat{\mathbf{y}} - \hat{\mathbf{X}}\boldsymbol{\beta}\|^2. \end{aligned} \tag{2.3}$$

Similarly, if the goal of the adversary is to increase the importance of the explanatory variable i, we can set our objective as

$$\min : -|\hat{\beta}_i| \tag{2.4}$$

withe the same constraints as in problem (2.3).

To solve the optimization problems (2.3) and (2.4), we first solve the following two optimization problems

$$\min_{\|[\mathbf{x}_0^\top, y_0]\| \le \eta} : \hat{\beta}_i \tag{2.5}$$

$$\text{s.t.} \quad \hat{\boldsymbol{\beta}} = \min_{\boldsymbol{\beta}} : \|\hat{\mathbf{y}} - \hat{\mathbf{X}}\boldsymbol{\beta}\|^2, \tag{2.6}$$

and

$$\max_{\|[\mathbf{x}_0^\top, y_0]\| \le \eta} : \hat{\beta}_i \tag{2.7}$$

$$\text{s.t.} \quad \hat{\boldsymbol{\beta}} = \min_{\boldsymbol{\beta}} : \|\hat{\mathbf{y}} - \hat{\mathbf{X}}\boldsymbol{\beta}\|^2. \tag{2.8}$$

It is easy to check that the solutions to problems (2.3) and (2.4) can be obtained from the solutions to problem (2.5) and (2.7). In particular, let $(\hat{\beta}_i^*)_{\min}$ and $(\hat{\beta}_i^*)_{\max}$ be optimal values of problem (2.5) and (2.7) respectively. Then, if $\hat{\beta}_i \ge 0$, we can check that $\max\{0, (\hat{\beta}_i^*)_{\min}\}$ and $\max\{|(\hat{\beta}_i^*)_{\min}|, |(\hat{\beta}_i^*)_{\max}|\}$ are the solutions to problem (2.3) and (2.4) respectively. Similar arguments can be made if $\hat{\beta}_i < 0$.

In the following, we will focus on solving the minimization problem (2.5). The solution to the maximization problem (2.7) can be obtained by using a similar approach. To solve this bi-level optimization problem, we can first solve the optimization problem in the subjective. Assume $\mathbf{X}$ is full column rank. Problem (2.6) is just an ordinary least squares problem, which has a simple closed-form solution: $\hat{\boldsymbol{\beta}} = (\hat{\mathbf{X}}^\top\hat{\mathbf{X}})^{-1}\hat{\mathbf{X}}^\top\hat{\mathbf{y}}$. Substitute in $\hat{\mathbf{X}} = [\mathbf{x}_0, \mathbf{X}^\top]^\top$ and $\hat{\mathbf{y}} = [y_0, \mathbf{y}^\top]^\top$, and we have

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X} + \mathbf{x}_0\mathbf{x}_0^\top)^{-1}[\mathbf{x}_0, \mathbf{X}^\top][y_0, \mathbf{y}^\top]^\top.$$

According to the Sherman-Morrison formula [78], we have

$$(\mathbf{X}^\top \mathbf{X} + \mathbf{x}_0 \mathbf{x}_0^\top)^{-1} = \mathbf{A} - \frac{\mathbf{A}\mathbf{x}_0 \mathbf{x}_0^\top \mathbf{A}}{1 + \mathbf{x}_0^\top \mathbf{A}\mathbf{x}_0}, \tag{2.9}$$

where

$$\mathbf{A} = (\mathbf{X}^\top \mathbf{X})^{-1}. \tag{2.10}$$

The inverse of $\mathbf{X}^\top \mathbf{X} + \mathbf{x}_0 \mathbf{x}_0^\top$ always exists because $1 + \mathbf{x}_0^\top \mathbf{A}\mathbf{x}_0 \neq 0$ and $\mathbf{X}^\top \mathbf{X}$ is invertible. Plug this inverse in the expression of $\hat{\boldsymbol{\beta}}$, we get

$$\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}_0 + \frac{\mathbf{A}\mathbf{x}_0 (y_0 - \mathbf{x}_0^\top \boldsymbol{\beta}_0)}{1 + \mathbf{x}_0^\top \mathbf{A}\mathbf{x}_0}, \tag{2.11}$$

where

$$\boldsymbol{\beta}_0 = \mathbf{A}\mathbf{X}^\top \mathbf{y}. \tag{2.12}$$

We can observe that $\boldsymbol{\beta}_0$ is the coefficient that is obtained from the clean data. Problem (2.5) is equivalent to

$$\begin{aligned} \min_{\mathbf{x}_0, y_0} &: \frac{\mathbf{a}^\top \mathbf{x}_0 (y_0 - \mathbf{x}_0^\top \boldsymbol{\beta}_0)}{1 + \mathbf{x}_0^\top \mathbf{A}\mathbf{x}_0} \\ \text{s.t.} \quad & \|[\mathbf{x}_0^\top, y_0]\| \leq \eta, \end{aligned} \tag{2.13}$$

where $\mathbf{a}$ is the ith column of $\mathbf{A}$. The optimization problem (2.13) is the ratio of two quadratic functions with a quadratic constraint. To further simplify this optimization problem, we can write our objective and subjective in a more compact form by performing variable change: $\mathbf{u} = [\mathbf{x}_0^\top, y_0]^\top$. Using this compact representation, the optimization problem (2.13) can be

written as

$$\begin{aligned} \min_{\mathbf{u}}: \quad & \frac{\frac{1}{2}\mathbf{u}^\top \mathbf{H}\mathbf{u}}{1+\mathbf{u}^\top \left[\begin{smallmatrix}\mathbf{A} & \mathbf{0}\\ \mathbf{0} & 0\end{smallmatrix}\right]\mathbf{u}} \\ \text{s.t.} \quad & \mathbf{u}^\top \mathbf{u} \leq \eta^2, \end{aligned} \tag{2.14}$$

in which

$$\mathbf{H} = \begin{bmatrix} -\mathbf{a}\boldsymbol{\beta}_0^\top - \boldsymbol{\beta}_0\mathbf{a}^\top & \mathbf{a} \\ \mathbf{a}^\top & 0 \end{bmatrix}. \tag{2.15}$$

(2.14) is a non-convex optimization problem. To solve this problem, we employ the technique introduced in [79]. We first perform variable change $\mathbf{u} = \frac{\mathbf{z}}{s}$ by introducing variable $\mathbf{z}$ and scalar s. Inserting this into problem (2.14), adding constraint 1 to the denominator of the objective and moving it to the subjective, we have a new optimization problem

$$\min_{\mathbf{z},s}: \quad \frac{1}{2}\mathbf{z}^\top \mathbf{H}\mathbf{z} \tag{2.16}$$

$$\text{s.t.} \quad s^2 + \mathbf{z}^\top \left[\begin{smallmatrix}\mathbf{A} & \mathbf{0}\\ \mathbf{0} & 0\end{smallmatrix}\right]\mathbf{z} = 1, \tag{2.17}$$

$$\mathbf{z}^\top \mathbf{z} \leq s^2\eta^2. \tag{2.18}$$

To validate the equivalence between problem (2.14) and (2.16), we only need to check if the optimal value of problem (2.14) is less than the optimal value of problem (2.16) when $s = 0$ [79]. Firstly, since $\mathbf{H}$ is not positive semi-definite (which will be shown later), the optimal value of problem (2.14) is less than zero. Secondly, when $s = 0$, the optimal value of problem (2.16) is zero, which is apparently larger than the optimal value of problem (2.14). Therefore, the two problems are equivalent.

To solve problem (2.16), we substitute s^2 in equation (2.17) for that in equation (2.18)

and obtain

$$\min_{\mathbf{z}}: \quad \frac{1}{2}\mathbf{z}^\top \mathbf{H}\mathbf{z} \tag{2.19}$$

$$\text{s.t.} \quad \frac{1}{2}\mathbf{z}^\top \mathbf{D}\mathbf{z} \leq \eta^2, \tag{2.20}$$

where

$$\mathbf{D} = 2\left(\mathbf{I} + \eta^2 \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}\right). \tag{2.21}$$

Notice that $\mathbf{H}$ is not positive semi-definite; hence problem (2.19) is not a standard convex QCQP problem [25]. However, it is proved that strong duality holds for this type of problem [80–82]. Hence, to solve this problem, we can start by investigating its KKT necessary conditions. The Lagrangian of problem (2.19) is

$$\mathcal{L}(\mathbf{z}, \lambda) = \frac{1}{2}\mathbf{z}^\top \mathbf{H}\mathbf{z} + \lambda\left(\frac{1}{2}\mathbf{z}^\top \mathbf{D}\mathbf{z} - \eta^2\right),$$

where λ is the dual variable. According to the KKT conditions, we have

$$(\mathbf{H} + \lambda\mathbf{D})\,\mathbf{z} = \mathbf{0}, \tag{2.22}$$

$$\frac{1}{2}\mathbf{z}^\top \mathbf{D}\mathbf{z} \leq \eta^2, \tag{2.23}$$

$$\lambda\left(\frac{1}{2}\mathbf{z}^\top \mathbf{D}\mathbf{z} - \eta^2\right) = 0, \tag{2.24}$$

$$\lambda \geq 0. \tag{2.25}$$

By inspecting the complementary slackness condition (2.24), we consider two cases based on the value of λ.

Case 1: $\lambda = 0$. In this case, we must have $\mathbf{H}\mathbf{z} = \mathbf{0}$ according to (2.22). As a result, the objective value of (2.19) is zero, which contradicts the fact that the optimal value should be

negative. Hence, this case is not possible.

Case 2: $\lambda > 0$. In this case, equality in (2.23) must hold based on (2.24). According to the stationary condition (2.22), if the matrix $\mathbf{H}+\lambda\mathbf{D}$ is full rank, we must have $\mathbf{z} = \mathbf{0}$, for which equality in (2.23) cannot hold. Hence, $\mathbf{H}+\lambda\mathbf{D}$ is not full-rank and we have $\det(\mathbf{H}+\lambda\mathbf{D}) = 0$. As $\mathbf{D}$ is positive definite, we also have $\det(\mathbf{D}^{-1/2}\mathbf{H}\mathbf{D}^{-1/2}+\lambda\mathbf{I}) = 0$. Since $\lambda > 0$, this equality tells us that $-\lambda$ belongs to one of the negative eigenvalues of $\mathbf{D}^{-1/2}\mathbf{H}\mathbf{D}^{-1/2}$. In the following, we will show that $\mathbf{D}^{-1/2}\mathbf{H}\mathbf{D}^{-1/2}$ has one and only one negative eigenvalue.

By definition, $\mathbf{D}$ is a block diagonal matrix. Hence, its inverse is also block diagonal. Let us define $\mathbf{D}^{-1/2} = \mathrm{diag}\{\mathbf{G}, g\}$, where $\mathbf{G} = 1/\sqrt{2}(\mathbf{I}+\eta^2\mathbf{A})^{-1/2}$ and $g = 1/\sqrt{2}$. Thus, we have

$$\mathbf{D}^{-1/2}\mathbf{H}\mathbf{D}^{-1/2} = \begin{bmatrix} -\mathbf{c}\mathbf{h}^\top - \mathbf{h}\mathbf{c}^\top & g\mathbf{c} \\ g\mathbf{c}^\top & 0 \end{bmatrix},$$

where $\mathbf{c} = \mathbf{G}\mathbf{a}$ and $\mathbf{h} = \mathbf{G}\boldsymbol{\beta}_0$. Define ξ as one eigenvalue of $\mathbf{D}^{-1/2}\mathbf{H}\mathbf{D}^{-1/2}$, and compute its eigenvalues by computing the characteristic polynomial:

$$\begin{aligned} &\det\left(\xi\mathbf{I} - \mathbf{D}^{-1/2}\mathbf{H}\mathbf{D}^{-1/2}\right) \\ &= \xi^{m-1}\left(\xi^2 + 2\xi\mathbf{c}^\top\mathbf{h} + \mathbf{c}^\top\mathbf{h}\mathbf{h}^\top\mathbf{c} - g^2\mathbf{c}^\top\mathbf{c} - \mathbf{c}^\top\mathbf{c}\mathbf{h}^\top\mathbf{h}\right). \end{aligned}$$

Thus, the eigenvalues of $\mathbf{D}^{-1/2}\mathbf{H}\mathbf{D}^{-1/2}$ are $\xi = 0$ ($(m-1)$ multiplicities) and $\xi = -\mathbf{c}^\top\mathbf{h} \pm \|\mathbf{c}\|\sqrt{g^2 + \mathbf{h}^\top\mathbf{h}}$. Since $\|\mathbf{c}\|\sqrt{g^2 + \mathbf{h}^\top\mathbf{h}} > |\mathbf{c}^\top\mathbf{h}|$, the eigenvalues of $\mathbf{D}^{-1/2}\mathbf{H}\mathbf{D}^{-1/2}$ satisfy: $\xi_{m+1} < 0, \quad \xi_m = \xi_{m-1} = \cdots = \xi_2 = 0, \quad \xi_1 > 0$. Now, it is clear that $\mathbf{D}^{-1/2}\mathbf{H}\mathbf{D}^{-1/2}$ has one and only one negative eigenvalue and one positive eigenvalue, respectively. Thus, we have $\lambda = -\xi_{m+1}$. Assume $\boldsymbol{\nu}_1$ and $\boldsymbol{\nu}_{m+1}$ are two eigenvectors corresponding to eigenvalues ξ_1 and ξ_{m+1}. Through simple calculation, we have

$$\boldsymbol{\nu}_i = k_i\left[-\frac{\mathbf{c}^\top\mathbf{h} + \xi_i}{\mathbf{c}^\top\mathbf{c}}\mathbf{c}^\top + \mathbf{h}^\top, \frac{g\mathbf{c}^\top}{\xi_i}\left(-\frac{\mathbf{c}^\top\mathbf{h} + \xi_i}{\mathbf{c}^\top\mathbf{c}}\mathbf{c} + \mathbf{h}\right)\right]^\top, \tag{2.26}$$

where $i = 1,\ m+1$ and scalar k_i is the normalization constant to guarantee the eigenvectors to be of unit length. According to (2.22), we have

$$(\mathbf{H} + \lambda \mathbf{D})\, \mathbf{z} = \mathbf{D}^{1/2} \left(\mathbf{D}^{-1/2} \mathbf{H} \mathbf{D}^{-1/2} + \lambda \mathbf{I} \right) \mathbf{D}^{1/2} \mathbf{z} = 0;$$

thus the solution to problem (2.19) is

$$\mathbf{z}^* = k \cdot \mathbf{D}^{-1/2} \boldsymbol{\nu}_{m+1}. \tag{2.27}$$

Since $\frac{1}{2} \mathbf{z}^\top \mathbf{D} \mathbf{z} = \eta^2$, we have $k = \sqrt{2}\eta$. Having the expression of the optimal $\mathbf{z}^*$, we can then compute s according to equation (2.17):

$$s = \pm \sqrt{1 - (\mathbf{z}^*_{1:m})^\top \mathbf{A}\, \mathbf{z}^*_{1:m}}, \tag{2.28}$$

where $\mathbf{z}^*_{1:m}$ is the vector that comprises the first m elements of $\mathbf{z}^*$. Hence, the corresponding solution to problem (2.13) is

$$\mathbf{x}_0^* = \mathbf{z}^*_{1:m}/s, \quad y_0^* = z^*_{m+1}/s. \tag{2.29}$$

We now compute the optimal value of problem (2.16). Since our objective function is $\frac{1}{2}(\mathbf{z}^*)^\top \mathbf{H} \mathbf{z}^*$, substituting $\mathbf{z}^*$ in (2.27) leads to the objective value: $\eta^2 \boldsymbol{\nu}_{m+1}^\top \mathbf{D}^{-1/2} \mathbf{H} \mathbf{D}^{-1/2} \boldsymbol{\nu}_{m+1}$. Since $\boldsymbol{\nu}_{m+1}^\top \mathbf{D}^{-1/2} \mathbf{H} \mathbf{D}^{-1/2} \boldsymbol{\nu}_{m+1} = \xi_{m+1}$, our optimal objective value is $\eta^2 \xi_{m+1}$.

Following similar analysis as above, we can find the optimal $\mathbf{z}^*$ for problem (2.7), which is $\mathbf{z}^* = \sqrt{2}\eta \mathbf{D}^{-1/2} \boldsymbol{\nu}_1$. Also, we can compute the optimal $\mathbf{x}_0^*$ and y_0^* according to equation (2.29) and its optimal objective value, which is $\eta^2 \xi_1$.

In summary, the optimal values for problems (2.5) and (2.7) are $\eta^2 \xi_{m+1} + (\boldsymbol{\beta}_0)_i$ and $\eta^2 \xi_1 + (\boldsymbol{\beta}_0)_i$ respectively. We have summarized the process to design the optimal adversarial data point in Algorithm 1 with respect to objective (2.5) and the process with respect to

Algorithm 1 Optimal Adversarial Data Point Design

1: **Input**: the data set, $\{y_i, \mathbf{x}_i\}_{i=1}^n$, energy budget η, and the index of feature to be attacked.
2: **Steps**:
3: compute $\mathbf{A}$ according to equation (2.10), compute $\boldsymbol{\beta}_0$ according to (2.12).
4: compute $\mathbf{H}$ and $\mathbf{D}$ according to (2.15) and (2.21), respectively.
5: compute the smallest eigenvalue, ξ_{m+1}, of $\mathbf{D}^{-1/2}\mathbf{H}\mathbf{D}^{-1/2}$ and its corresponding eigenvector according to (2.26).
6: design the adversarial data point, $\{\mathbf{x}_0, y_0\}$, according to equations (2.27), (2.28), and (2.29).
7: **Output**: return the optimal adversarial data point $\{\mathbf{x}_0, y_0\}$ and the optimal value $\eta^2\xi_{m+1} + (\boldsymbol{\beta}_0)_i$.

objective (2.7) can be obtained accordingly. Based on our optimal values of problems (2.5) and (2.7), we can further decide the optimal values of problems (2.3) and (2.4) as discussed at the beginning of this section. From our analysis we can see that the main computation is to compute $\mathbf{A}$ in (2.10). Hence, the complexity of our algorithm is $\mathcal{O}(m^3)$.

Moreover, if we use the ridge regression method in linear regression, there is only a slight difference in the matrix $\mathbf{A}$ in problem (2.13) and the whole analysis remains the same.

One may concern that the proposed adversarial data point may behave as an outlier and can be easily detected by the learning system. We can mitigate this by a simple repeating strategy, in which we repeat the proposed adversarial data point K times and shrink the magnitude of these poisoning data by $\sqrt{K}$. This can be simply verified by

$$\begin{aligned}
\hat{\boldsymbol{\beta}} &= (\hat{\mathbf{X}}^\top\hat{\mathbf{X}})^{-1}\hat{\mathbf{X}}\hat{\mathbf{y}} \\
&= \left(\mathbf{X}^\top\mathbf{X} + \mathbf{x}_0\mathbf{x}_0^\top\right)^{-1}\left(\mathbf{X}^\top\mathbf{y} + \mathbf{x}_0 y_0\right) \\
&= \left(\mathbf{X}^\top\mathbf{X} + \sum_{i=1}^{k}\frac{1}{\sqrt{K}}\mathbf{x}_0\frac{1}{\sqrt{K}}\mathbf{x}_0^\top\right)^{-1}\Big(\mathbf{X}^\top\mathbf{y} \\
&\quad + \sum_{i=1}^{K}\frac{1}{\sqrt{K}}\mathbf{x}_0\frac{1}{\sqrt{K}}y_0\Big) \\
&= (\tilde{\mathbf{X}}^\top\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}^\top\tilde{\mathbf{y}},
\end{aligned}$$

where $\tilde{\mathbf{X}} = [\mathbf{X}^\top, \underbrace{\frac{1}{\sqrt{K}}\mathbf{x}_0, \ldots, \frac{1}{\sqrt{K}}\mathbf{x}_0}_{K \text{ times}}]^\top$ and $\tilde{\mathbf{y}} = [\mathbf{y}^\top, \underbrace{\frac{1}{\sqrt{K}}y_0, \ldots, \frac{1}{\sqrt{K}y_0}}_{K \text{ times}}]^\top$. By shrinking the poisoning data points, it will make the detection of these points more difficult, especially when the dataset is standardized.

We now analyze the impact of parameters, such as η, on the objective value. Even though we have a closed-form solution to the optimal adversarial data point, the objective is a complex function of the original dataset. Hence, it will be difficult to analyze this for the general case. Instead, we will focus on some special cases. In particular, we analyze how the energy budget affects the value of objective function in the large data sample scenario. As our analysis shows, our optimal values are $\eta^2\xi$, where $\xi = -\mathbf{c}^\top\mathbf{h} \pm \|\mathbf{c}\|\sqrt{g^2 + \mathbf{h}^\top\mathbf{h}}$, $\mathbf{c} = \mathbf{G}\mathbf{a}$, $\mathbf{h} = \mathbf{G}\boldsymbol{\beta}_0$, $\mathbf{G} = 1/\sqrt{2}(\mathbf{I} + \eta^2\mathbf{A})^{-1/2}$, $g = 1/\sqrt{2}$, $\mathbf{A} = (\mathbf{X}^\top\mathbf{X})^{-1}$, and $\boldsymbol{\beta}_0$ is the original regression coefficient. In the large data sample limit and the assumption that the features are independent and standardized, we have the approximation $\mathbf{A} = \mathbf{I}$. Recall that $\mathbf{a}$ is the ith column of $\mathbf{A}$, $\mathbf{a} = \mathbf{e}_i$. As the result, the objective value is $\eta^2\xi = \frac{1}{2}\frac{\eta^2}{1+\eta^2}\left[-\boldsymbol{\beta}_0^i \pm \sqrt{\eta^2 + 1 + ||\boldsymbol{\beta}_0||^2}\right]$. For objective (2.5) with optimal value $\frac{1}{2}\frac{\eta^2}{1+\eta^2}\left[-\boldsymbol{\beta}_0^i - \sqrt{\eta^2 + 1 + ||\boldsymbol{\beta}_0||^2}\right]$, this function is monotonically decreasing with η. For the objective (2.7) with optimal value $\frac{1}{2}\frac{\eta^2}{1+\eta^2}\left[-\boldsymbol{\beta}_0^i + \sqrt{\eta^2 + 1 + ||\boldsymbol{\beta}_0||^2}\right]$, it is a monotonically increasing function of η.

2.2.3 Attacking with Small Changes of Other Regression Coefficients

In Chapter 2.2.2, we have discussed how to design the adversarial data points to attack one specific regression coefficient. However, as we only focus on one particular regression coefficient, other regression coefficients may also be changed. In this subsection, we consider a more complex objective function, where we aim to make the changes to other regression coefficients to be as small as possible while attacking one of the regression coefficients.

Suppose our objective is to minimize the ith regression coefficient (the scenario of max-

imize the ith regression coefficient can be solved using similar approach), i.e., to minimize $\|\hat{\beta}_i\|^2$. At the same time, we would also like to minimize the changes to the rest of the regression coefficients, i.e., to minimize $\|\boldsymbol{\beta}_0^{-i} - \hat{\boldsymbol{\beta}}^{-i}\|^2$, where $\boldsymbol{\beta}_0^{-i} = [\beta_0^1, \ldots, \beta_0^{i-1}, 0, \beta_0^{i+1}, \ldots, \beta_0^m]^\top$ and $\hat{\boldsymbol{\beta}}^{-i} = [\hat{\beta}_1, \ldots, \hat{\beta}_{i-1}, 0, \hat{\beta}_{i+1}, \hat{\beta}_m]^\top$. Combine the two objectives, we have our new objective function

$$f(\hat{\boldsymbol{\beta}}) = \frac{1}{2}\left\|\boldsymbol{\beta}_0^{-i} - \hat{\boldsymbol{\beta}}^{-i}\right\|^2 + \frac{\lambda}{2}\left\|\hat{\beta}_i\right\|^2,$$

where λ is the trade-off parameter. The larger the λ is, the more effort will be made to keep the ith regression coefficient small. A negative λ means the adversary attempts to make the magnitude of the ith regression coefficient large. Again, we assume that the attack energy budget is η. As the result, we have the following optimization problem

$$\begin{aligned} \min_{\left\|[\mathbf{x}_0^\top, y_0]\right\| \leq \eta} : \quad & \frac{1}{2}\left\|\boldsymbol{\beta}_0^{-i} - \hat{\boldsymbol{\beta}}^{-i}\right\|^2 + \frac{\lambda}{2}\left\|\hat{\beta}_i\right\|^2 \\ \text{s.t.} \quad & \hat{\boldsymbol{\beta}} = \operatorname*{argmin}_{\boldsymbol{\beta}} : \|\hat{\mathbf{y}} - \hat{\mathbf{X}}\boldsymbol{\beta}\|^2. \end{aligned} \tag{2.30}$$

As the objective function is a quadratic function with respect to $\hat{\boldsymbol{\beta}}$, we can write it in a more compact form: $\frac{1}{2}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0^{-i})^\top \boldsymbol{\Lambda}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0^{-i})$, where $\boldsymbol{\Lambda} = \operatorname{diag}(1, 1, \ldots, \lambda, \ldots, 1)$ and λ is at the ith coordinate. With this compact form, our optimization problem can be written as

$$\begin{aligned} \min_{\left\|[\mathbf{x}_0^\top, y_0]\right\| \leq \eta} : \quad & \frac{1}{2}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0^{-i})^\top \boldsymbol{\Lambda}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0^{-i}) \\ \text{s.t.} \quad & \hat{\boldsymbol{\beta}} = \operatorname*{argmin}_{\boldsymbol{\beta}} : \|\hat{\mathbf{y}} - \hat{\mathbf{X}}\boldsymbol{\beta}\|^2. \end{aligned} \tag{2.31}$$

To solve this problem, same as in the previous subsection, we start by solving the lower level optimization problem. Since we have the same lower level problem as in (2.5), substitute $\hat{\boldsymbol{\beta}}$

Algorithm 2 Optimal Adversarial Data Point Design while Making Small Changes to Other Regression Coefficients

1: **Input**: the data set, $\{y_i, \mathbf{x}_i\}_{i=1}^n$, energy budget η, and the index of feature to be attacked, the trade-off parameter λ.
2: **Steps**:
3: compute $\mathbf{A}$ according to equation (2.10), compute $\boldsymbol{\beta}_0$ according to (2.12), compute $\mathbf{A}_2$ according to (2.32).
4: follow the steps (2.30), (2.31), (2.33), and (2.34), and formulate our problem as a polynomial optimization problem (2.37).
5: use Lasserre's relaxation method to solve problem (2.37) and get the optimal solution $\mathbf{x}^*$ and optimal value p^*.
6: compute $\mathbf{w}^* = \mathbf{U}^\top \mathbf{x}^*$, where $\mathbf{I} + \eta^2 \mathbf{A}_2 = \mathbf{U}\mathbf{U}^\top$.
7: compute $s^* = \pm\sqrt{1 - (\mathbf{w}^*)^\top \mathbf{A}_2 \mathbf{w}^*}$.
8: calculate the optimal solution $\mathbf{x}_0^* = \mathbf{w}_{1:m}^*/s^*, y_0^* = w_{m+1}^*/s^*$.
9: **Output**: return the optimal adversarial data point $\{y_0^*, \mathbf{x}_0^*\}$ and the optimal value p^*.

in the objective with the expression (2.11), and we have the one level optimization problem

$$\begin{aligned} \min_{\mathbf{x}_0, y_0} : \quad & \frac{1}{2}\mathbf{g}^\top \boldsymbol{\Lambda} \mathbf{g} \\ \text{s.t.} \quad & \left\| [\mathbf{x}_0^\top, y_0] \right\| \leq \eta, \end{aligned}$$

where $\mathbf{g} = \frac{\mathbf{A}\mathbf{x}_0(y_0 - \mathbf{x}_0^\top \boldsymbol{\beta}_0)}{1 + \mathbf{x}_0^\top \mathbf{A} \mathbf{x}_0} - \mathbf{b}$ with $\mathbf{A}$ and $\boldsymbol{\beta}_0$ defined in (2.10) and (2.12) respectively and $\mathbf{b} = \boldsymbol{\beta}_0^{-i} - \boldsymbol{\beta}_0$. To further simplify our problem, let us define

$$\mathbf{A}_1 = [\mathbf{A}, \mathbf{0}],\ \mathbf{A}_2 = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix},\ \mathbf{c} = \begin{bmatrix} -\boldsymbol{\beta}_0 \\ 1 \end{bmatrix},\ \mathbf{z} = \begin{bmatrix} \mathbf{x}_0 \\ y_0 \end{bmatrix}, \tag{2.32}$$

where $\mathbf{A}_1 \in \mathbb{R}^{m \times (m+1)}$ and $\mathbf{A}_2 \in \mathbb{R}^{(m+1)\times(m+1)}$. With the new defined variables, we can write our problem more compactly as:

$$\begin{aligned} \min_{\mathbf{z}} : \quad & \frac{1}{2}\left(\frac{\mathbf{A}_1 \mathbf{z}\mathbf{c}^\top \mathbf{z}}{1 + \mathbf{z}^\top \mathbf{A}_2 \mathbf{z}} - \mathbf{b} \right)^\top \boldsymbol{\Lambda} \left(\frac{\mathbf{A}_1 \mathbf{z}\mathbf{c}^\top \mathbf{z}}{1 + \mathbf{z}^\top \mathbf{A}_2 \mathbf{z}} - \mathbf{b} \right) \\ \text{s.t.} \quad & \|\mathbf{z}\| \leq \eta. \end{aligned} \tag{2.33}$$

Since the objective is a ratio of two quartic functions, similar to the process we carried out from (2.14) to (2.16), we perform variable change $\mathbf{z} = \frac{\mathbf{w}}{s}$ by introducing the new variable $\mathbf{w}$ and scalar s. Insert it into problem (2.33) and follow the same argument we have made to transform problem (2.14) to problem (2.16), problem (2.33) is equivalent to the following problem

$$\min_{\mathbf{w},s}: \quad \frac{1}{2}\left(\mathbf{A}_1\mathbf{w}\mathbf{c}^\top\mathbf{w} - \mathbf{b}\right)^\top \mathbf{\Lambda}\left(\mathbf{A}_1\mathbf{w}\mathbf{c}^\top\mathbf{w} - \mathbf{b}\right) \tag{2.34}$$

$$\text{s.t.} \quad (s^2 + \mathbf{w}^\top\mathbf{A}_2\mathbf{w})^2 = 1, \tag{2.35}$$

$$\mathbf{w}^\top\mathbf{w} \leq s^2\eta^2. \tag{2.36}$$

According to the definition of $\mathbf{A}_2$, it is positive semidefinite. Hence, we have $s^2 = 1 - \mathbf{w}^\top\mathbf{A}_2\mathbf{w}$. Plug in the expression of s^2 into (2.36), the constraints in problem (2.34) can be simplified to $\mathbf{w}^\top(\mathbf{I} + \eta^2\mathbf{A}_2)\mathbf{w} \leq \eta^2$. Let $\mathbf{U}^\top\mathbf{U} = \mathbf{I} + \eta^2\mathbf{A}_2$ be the Cholesky decomposition of $\mathbf{I} + \eta^2\mathbf{A}_2$. Define $\mathbf{H} = \mathbf{A}_1\mathbf{U}^{-1}$, $\mathbf{e} = \mathbf{U}^{-\top}\mathbf{c}$, and $\mathbf{x} = \mathbf{U}\mathbf{w}$, we can simplify problem (2.34) further as:

$$\min_{\mathbf{x}}: \quad \frac{1}{2}\left(\mathbf{H}\mathbf{x}\mathbf{e}^\top\mathbf{x} - \mathbf{b}\right)^\top \mathbf{\Lambda}\left(\mathbf{H}\mathbf{x}\mathbf{e}^\top\mathbf{x} - \mathbf{b}\right) \tag{2.37}$$

$$\text{s.t.} \quad \mathbf{x}^\top\mathbf{x} \leq \eta^2.$$

This is an optimization problem with a quartic objective function and with a quadratic constraint. Recent progress in multivariate polynomial optimization has made it possible to solve this problem using the sum of squares technology [83–86]. This method finds the globally optimal solutions by solving a sequence of convex linear matrix inequality problems. Even though this sequence might be infinitely long, in practice, a very short sequence is enough to guarantee its global optimality. Hence, in this subsection, we will resort to Lasserre's relaxation method [83]. Algorithm 2 summarizes the process to design the adversarial data point. The complexity of Algorithm 2 is dominant by the solving of the relaxation semidefinite problem. Hence, the computational complexity of Algorithm 2 is $\mathcal{O}(s(N)^{4.5})$,

Table 2.1: Configurations of $\mathbf{c}$ and $\mathbf{d}$ and their corresponding modifications.

Modification	Configurations of $\mathbf{c}$ and $\mathbf{d}$
delete the ith data sample	$\mathbf{c} = -\mathbf{e}_i$, $\mathbf{b} = \mathbf{X}_{i,:}$
delete feature i	$\mathbf{c} = \mathbf{X}_{:,i}^\top$, $\mathbf{d} = -\mathbf{e}_i$
add one adversarial data sample	$\mathbf{X} \leftarrow [\mathbf{X}, \mathbf{0}]$, $\mathbf{c} = \mathbf{e}_{n+1}$, $\mathbf{d} = \mathbf{x}_{n+1}^\top$
modify one entry	$\mathbf{c} = \eta \cdot \mathbf{e}_i$, $\mathbf{d} = \mathbf{e}_j$

where N is the relaxation order and $s(N) = \binom{N+m}{N}$ [87]. Numerical examples using this method to solve our problem with real data will be provided in Chapter 2.4.

In this subsection, we put an ℓ_2 norm constraint on the adversarial data point. It is possible to extend our work to other kinds of norm constraints, such as ℓ_1 and ℓ_∞ norm constraints. Suppose we put ℓ_p ($p = 1$ or $p = \infty$) norm constraint on the adversarial data sample with objective (2.30), following similar steps in this subsection, we can obtain objective (2.34) with constraint (2.35) and the norm cone constraint $||\mathbf{w}||_p \leq s\eta$. When $p = 1$, the norm cone constraint can be transformed to the inequalities constraints $\sum_{i=1}^{m+1} a_i \leq s\eta$ and $-a_i \leq w_i \leq a_i$ for $i = 1, \ldots, m+1$, where a_i is the auxiliary variable. When $p = \infty$, we can transform the norm cone constraint to $b \leq s\eta$ and $-b\mathbf{1} \preccurlyeq \mathbf{w} \preccurlyeq b\mathbf{1}$, where b is a auxiliary variable. Both cases lead to linear inequality constraints, which are special polynomial inequalities. Hence, we can still use the Lasserre's relaxation method to obtain the optimal solution.

2.3 Rank-one Attack Analysis

In Chapter 2.2, we have discussed how to design one adversarial data point to attack the regression coefficients. In this section, we consider a more powerful adversary who can modify the whole dataset in order to attack the regression coefficients. In particular, we will consider a rank-one attack on the feature matrix [26]. This type of attack covers many practical scenarios, for example, modifying one entry of the feature matrix, deleting one feature, changing one feature, replacing one feature, etc. We summarize the these modifications

and their corresponding configurations of $\mathbf{c}$ and $\mathbf{d}$ in Table 2.1, where $\mathbf{cd}^\top$ is the rank one modification matrix, $\mathbf{X}_{i,:}$ denotes the ith row of the feature matrix $\mathbf{X}$, $\mathbf{X}_{:,i}$ represents the ith column of the feature matrix, $\mathbf{e}_i$ is the standard basis vector, and η is the scalar which denotes the modification energy budget. Hence, the analysis of the rank-one attack provides a universal bound for all of these kinds of modifications. Specifically, we will consider the objective in problem (2.3) and (2.4) where the adversary attacks one particular regression coefficient. In the following, we will first formulate our problem and then provide our alternating optimization method to solve this problem.

In the considered rank-one attack model, the attacker will carefully design a rank-one feature modification matrix $\mathbf{\Delta}$ and add it to the original feature matrix $\mathbf{X}$. As the result, the modified feature matrix is $\hat{\mathbf{X}} = \mathbf{X} + \mathbf{\Delta}$. As $\mathbf{\Delta}$ has rank one, we can write $\mathbf{\Delta} = \mathbf{cd}^\top$, where $\mathbf{c} \in \mathbb{R}^n$ and $\mathbf{d} \in \mathbb{R}^m$. Similar to the previous section, we restrict the adversary to having a limited energy budget, η. Here, we use the Frobenius norm to measure the energy of the modification matrix. Hence, we have $\|\mathbf{\Delta}\|_{\mathrm{F}} \leq \eta$, where $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm of a matrix. If the attacker's goal is to increase the importance of feature i, our problem can be written as

$$\begin{aligned} \max_{\|\mathbf{cd}^\top\|_{\mathrm{F}} \leq \eta} : \quad & |\hat{\beta}_i| \\ \text{s.t.} \quad & \hat{\boldsymbol{\beta}} = \operatorname*{argmin}_{\boldsymbol{\beta}} \|\mathbf{y} - \hat{\mathbf{X}}\boldsymbol{\beta}\|^2, \\ & \hat{\mathbf{X}} = \mathbf{X} + \mathbf{cd}^\top. \end{aligned} \tag{2.38}$$

If the adversary is trying to minimize the magnitude of the ith regression coefficient, our

problem is

$$\begin{aligned} \min_{\|\mathbf{c}\mathbf{d}^\top\|_{\mathrm{F}} \le \eta} : \quad & |\beta_i| \\ \text{s.t.} \quad & \hat{\boldsymbol{\beta}} = \operatorname*{argmin}_{\boldsymbol{\beta}} : \ \|\mathbf{y} - \hat{\mathbf{X}}\boldsymbol{\beta}\|^2, \\ & \hat{\mathbf{X}} = \mathbf{X} + \mathbf{c}\mathbf{d}^\top. \end{aligned} \tag{2.39}$$

Similar as in Chapter 2.2.2, the solutions to problems (2.38) and (2.39) can be obtained by the solutions to the following two problems:

$$\max_{\|\mathbf{c}\mathbf{d}^\top\|_{\mathrm{F}} \le \eta} : \quad \hat{\beta}_i \tag{2.40}$$

and

$$\min_{\|\mathbf{c}\mathbf{d}^\top\|_{\mathrm{F}} \le \eta} : \quad \hat{\beta}_i \tag{2.41}$$

with the same constraints as in (2.38) and (2.39).

We can further write the above two problems in a more unified form:

$$\begin{aligned} \min_{\|\mathbf{c}\mathbf{d}^\top\|_{\mathrm{F}} \le \eta} : \quad & \mathbf{e}^\top \hat{\boldsymbol{\beta}} \\ \text{s.t.} \quad & \hat{\boldsymbol{\beta}} = \operatorname*{argmin}_{\boldsymbol{\beta}} : \ \|\mathbf{y} - \hat{\mathbf{X}}\boldsymbol{\beta}\|^2, \\ & \hat{\mathbf{X}} = \mathbf{X} + \mathbf{c}\mathbf{d}^\top. \end{aligned} \tag{2.42}$$

If $\mathbf{e} = \mathbf{e}_i$, in which $\mathbf{e}_i$ is a vector with the ith entry being 1 and all other entries being zero, problem (2.42) is equivalent to problem (2.41). If $\mathbf{e} = -\mathbf{e}_i$, problem (2.42) is equivalent to problem (2.40). Hence, in the following part, we will focus on solving this unified problem (2.42).

To solve problem (2.42), we can first solve the lower level optimization problem in the

constraints. It admits a simple solution that $\hat{\boldsymbol{\beta}} = \hat{\mathbf{X}}^{\dagger}\mathbf{y}$ and $\hat{\mathbf{X}}^{\dagger}$ is the pseudo-inverse of $\hat{\mathbf{X}}$. This pseudo-inverse can be written as $\hat{\mathbf{X}}^{\dagger} = \mathbf{X}^{\dagger} + \mathbf{G}$ [88], where

$$\mathbf{G} = \frac{1}{\gamma}\mathbf{X}^{\dagger}\mathbf{n}\mathbf{w}^{\top} - \frac{\gamma}{\|\mathbf{n}\|^2\|\mathbf{w}\|^2 + \gamma^2} \cdot \left(\frac{\|\mathbf{w}\|^2}{\gamma}\mathbf{X}^{\dagger}\mathbf{n} + \mathbf{v}\right)\left(\frac{\|\mathbf{n}\|^2}{\gamma}\mathbf{w} + \mathbf{n}\right)^{\top}, \tag{2.43}$$

$\gamma = 1 + \mathbf{d}^{\top}\mathbf{X}^{\dagger}\mathbf{c}$, $\mathbf{v} = \mathbf{X}^{\dagger}\mathbf{c}$, $\mathbf{n} = (\mathbf{X}^{\dagger})^{\top}\mathbf{d}$, and $\mathbf{w} = (\mathbf{I} - \mathbf{X}\mathbf{X}^{\dagger})\mathbf{c}$.

Since $\hat{\boldsymbol{\beta}} = \hat{\mathbf{X}}^{\dagger}\mathbf{y} = (\mathbf{X}^{\dagger} + \mathbf{G})\mathbf{y}$ and $\mathbf{X}^{\dagger}$ does not depend on $\mathbf{c}$ and $\mathbf{d}$, our problem is equivalent to

$$\begin{aligned} \min_{\mathbf{c},\mathbf{d}}: \quad & \mathbf{e}^{\top}\mathbf{G}\mathbf{y} \\ \text{s.t.} \quad & \|\mathbf{c} \cdot \mathbf{d}^{\top}\|_{\mathrm{F}} \leq \eta. \end{aligned} \tag{2.44}$$

Suppose $(\mathbf{c}^*, \mathbf{d}^*)$ is the optimal solution of (2.44), it is easy to see that for nonzero k, $(k\mathbf{c}^*, \mathbf{d}^*/k)$ is also a valid optimal solution. To avoid the ambiguity, it is necessary and possible to reduce the feasible region further. Hence, we put an extra constraint on $\mathbf{c}$, where we restrict the norm of $\mathbf{c}$ to be less than or equal to 1. As a result, our problem can be further written as

$$\begin{aligned} \min_{\mathbf{c},\mathbf{d}}: \quad & \mathbf{e}^{\top}\mathbf{G}\mathbf{y} \\ \text{s.t.} \quad & \|\mathbf{c}\| \leq 1, \quad \|\mathbf{d}\| \leq \eta, \end{aligned} \tag{2.45}$$

in which we use the identity $\|\mathbf{c}\mathbf{d}^{\top}\|_{\mathrm{F}} = \|\mathbf{c}\|\|\mathbf{d}\|$. It is clear that problem (2.44) and problem (2.45) have the same optimal objective value.

Since $\mathbf{G}$ is determined by $\mathbf{c}$, $\mathbf{d}$, and $\mathbf{X}$, different values of $\mathbf{c}$ and $\mathbf{d}$ may result in different objective functions. Before further discussion, let us assume the singular value decomposition of the original feature matrix is $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\top}$, where $\boldsymbol{\Sigma} = [\mathrm{diag}(\sigma_1, \sigma_2, \cdots, \sigma_m), \mathbf{0}]^{\top}$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m > 0$. With this decomposition, we have $\mathbf{X}^{\dagger} = \mathbf{V}\boldsymbol{\Sigma}^{\dagger}\mathbf{U}^{\top}$, where $\boldsymbol{\Sigma}^{\dagger} = [\mathrm{diag}(\sigma_1^{-1}, \sigma_2^{-1}, \cdots, \sigma_m^{-1}), \mathbf{0}]$. In (2.43), if $\eta \geq \sigma_m$, by letting $\gamma \to 0$, we have our objective

being minus infinity by setting $(\mathbf{c}, \mathbf{d}) = (\mathbf{u}_m, -\sigma_m \mathbf{v}_m)$ or $(\mathbf{c}, \mathbf{d}) = (-\mathbf{u}_m, \sigma_m \mathbf{v}_m)$, where $\mathbf{u}_m$ and $\mathbf{v}_m$ are the mth column of matrices $\mathbf{U}$ and $\mathbf{V}$, respectively. Hence, we conclude that, when $\eta \geq \sigma_m$, the optimal value of problem (2.45) is unbounded from below. As the result, throughout this section, we assume $\eta < \sigma_m$. Thus, we also have $\gamma = 1 + \mathbf{d}^\top \mathbf{X}^\dagger \mathbf{c} \geq 1 - \|\mathbf{c} \cdot \mathbf{d}^\top\| \|\mathbf{X}^\dagger\| \geq 1 - \frac{\eta}{\sigma_m} > 0$. We note that when η approaches σ_m, it does not mean to kill all of the signals in the feature matrix but only some signals with the energy equal to the smallest singular value of the feature matrix.

Let h denote our objective $h(\mathbf{c}, \mathbf{d}) = \mathbf{e}^\top \mathbf{G} \mathbf{y}$, plug in the expression of $\mathbf{G}$, and we have

$$\begin{aligned} h(\mathbf{c}, \mathbf{d}) = & \frac{1}{\|\mathbf{n}\|^2 \|\mathbf{w}\|^2 + \gamma^2} \big(\gamma \mathbf{e}^\top \mathbf{X}^\dagger \mathbf{n} \mathbf{w}^\top \mathbf{y} - \gamma \mathbf{e}^\top \mathbf{v} \mathbf{n}^\top \mathbf{y} \\ & - \|\mathbf{w}\|^2 \mathbf{e}^\top \mathbf{X}^\dagger \mathbf{n} \mathbf{n}^\top \mathbf{y} - \|\mathbf{n}\|^2 \mathbf{e}^\top \mathbf{v} \mathbf{w}^\top \mathbf{y} \big). \end{aligned} \tag{2.46}$$

We need to optimize $h(\mathbf{c}, \mathbf{d})$ over $\mathbf{c}$ and $\mathbf{d}$ with the constraint $\|\mathbf{c}\| \leq 1$ and $\|\mathbf{d}\| \leq \eta$. However, $h(\mathbf{c}, \mathbf{d})$ is a ratio of two quartic functions, which is known to be a hard non-convex problem in general. To solve this problem, similar to [34], we can use the projected gradient descent method. However, it is hard to choose a proper step-size and its convergence is not clear when the projected gradient descent is applied to a non-convex problem. In the following, we provide an alternating optimization algorithm with provable convergence.

The enabling observation of our approach is that even though the optimization problem is a complex non-convex problem, for a fixed $\mathbf{c}$, h is a ratio of two quadratic functions with respect to $\mathbf{d}$. Similarly, for a fixed $\mathbf{d}$, h is a ratio of two quadratic functions with respect to $\mathbf{c}$. A ratio of two quadratic functions admits a hidden convex structure [89]. Inspired by this, we decompose our optimization variables into $\mathbf{c}$ and $\mathbf{d}$, and then use alternating optimization algorithm described in Algorithm 3 to sequentially optimize $\mathbf{c}$ and $\mathbf{d}$.

The core of this algorithm is to solve the following two problems

$$\mathbf{c}^k = \underset{\|\mathbf{c}\| \leq 1}{\operatorname{argmin}} : h(\mathbf{c}, \mathbf{d}^{k-1}), \tag{2.47}$$

Algorithm 3 Optimal Rank-one Attack Matrix Design via the Alternating Optimization Algorithm

Input: data set $\{y_i, \mathbf{x}_i\}_{i=1}^n$ and energy budget η.
Initialize: randomly initialize $\mathbf{c}^0$ and $\mathbf{d}^0$, set number of iterations $k = 0$.
compute $\mathbf{G}$ according to (2.43).
plug in the expression of $\mathbf{G}$ into (2.45), and obtain our objective, $h(\mathbf{c}, \mathbf{d})$, as in (2.46).
Do
update $\mathbf{c}^k$ by solving: $\mathbf{c}^k = \underset{\|\mathbf{c}\| \leq 1}{\operatorname{argmin}} : h(\mathbf{c}, \mathbf{d}^{k-1})$,
update $\mathbf{d}^k$ by solving: $\mathbf{d}^k = \underset{\|\mathbf{d}\| \leq \eta}{\operatorname{argmin}} : h(\mathbf{c}^k, \mathbf{d})$,
set $k = k + 1$,
While convergence conditions are not meet.
compute the modification matrix $\mathbf{\Delta} = \mathbf{c}^k (\mathbf{d}^k)^\top$.
Output: return the modification matrix, $\mathbf{\Delta}$.

and

$$\mathbf{d}^k = \underset{\|\mathbf{d}\| \leq \eta}{\operatorname{argmin}} : h(\mathbf{c}^k, \mathbf{d}). \tag{2.48}$$

For a fixed $\mathbf{d}$, the objective of problem (2.47) becomes $h(\mathbf{c}, \mathbf{d}) = h_1(\mathbf{c})/h_2(\mathbf{c})$, where we omit the superscript of $\mathbf{d}$,

$$\begin{aligned} h_1(\mathbf{c}) = \mathbf{c}^\top \big[& \mathbf{e}^\top \mathbf{X}^\dagger \mathbf{n}\mathbf{n}\mathbf{y}^\top (\mathbf{I} - \mathbf{X}\mathbf{X}^\dagger) - \mathbf{n}^\top \mathbf{y} \mathbf{n} \mathbf{e}^\top \mathbf{X}^\dagger \\ & - \mathbf{e}^\top \mathbf{X}^\dagger \mathbf{n}\mathbf{n}^\top \mathbf{y} (\mathbf{I} - \mathbf{X}\mathbf{X}^\dagger) - \|\mathbf{n}\|^2 (\mathbf{X}^\dagger)^\top \mathbf{e}\mathbf{y}^\top (\mathbf{I} - \mathbf{X}\mathbf{X}^\dagger) \big] \mathbf{c} \\ & + \big[\mathbf{e}^\top \mathbf{X}^\dagger \mathbf{n} (\mathbf{I} - \mathbf{X}\mathbf{X}^\dagger) \mathbf{y} - \mathbf{n}^\top \mathbf{y} (\mathbf{X}^\dagger)^\top \mathbf{e} \big]^\top \mathbf{c}, \end{aligned} \tag{2.49}$$

and

$$h_2(\mathbf{c}) = \mathbf{c}^\top \big[\|\mathbf{n}\|^2 (\mathbf{I} - \mathbf{X}\mathbf{X}^\dagger) + \mathbf{n}\mathbf{n}^\top \big] \mathbf{c} + 2\mathbf{n}^\top \mathbf{c} + 1. \tag{2.50}$$

Hence, problem (2.47) can be written as:

$$\min_{\mathbf{c}} : \quad \frac{h_1(\mathbf{c})}{h_2(\mathbf{c})} \tag{2.51}$$

$$\text{s.t.} \quad \|\mathbf{c}\| \leq 1, \tag{2.52}$$

where the forms of $h_i(\mathbf{c}) = \mathbf{c}^\top \mathbf{A}_i \mathbf{c} + 2\mathbf{b}_i^\top \mathbf{c} + l_i$, $i = 1, 2$ and $\mathbf{A}_i$, $\mathbf{b}_i$ and l_i can be derived from (2.49) and (2.50). The objective of this problem is the ration of two quadratic functions. Even though it is non-convex, it has certain hidden convex structures. The following theorem characterizes its optimal solution by solving a semidefinite programming [89].

Theorem 2.1. *([89]) If there exists $\mu > 0$ such that*

$$\begin{bmatrix} \mathbf{A}_2 & \mathbf{b}_2 \\ \mathbf{b}_2^\top & l_2 \end{bmatrix} + \mu \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -1 \end{bmatrix} \succ \mathbf{0}, \tag{2.53}$$

the optimal value of problem (2.51) *is equivalent to the following optimal value*

$$\begin{aligned} \max_{\alpha, \nu \geq 0} : \quad & \alpha \\ s.t. \quad & \begin{bmatrix} \mathbf{A}_1 & \mathbf{b}_1 \\ \mathbf{b}_1^\top & l_1 \end{bmatrix} \succeq \alpha \begin{bmatrix} \mathbf{A}_2 & \mathbf{b}_2 \\ \mathbf{b}_2^\top & l_2 \end{bmatrix} - \nu \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -1 \end{bmatrix} \end{aligned} \tag{2.54}$$

Proof. Please see [89] for detail. □

We now show that our problem (2.51) satisfies condition (2.53). As the result, we can find the solution to problem (2.51) by solving problem (2.54).

To prove the left hand side of (2.53) is positive definite, we can show the following two

inequalities are true according to Schur complement condition for positive definite matrix

$$l_2 - \mu > 0, \tag{2.55}$$

$$\mathbf{A}_2 + \mu\mathbf{I} - \frac{1}{1-\mu}\mathbf{b}_2\mathbf{b}_2^\top \succ \mathbf{0}, \tag{2.56}$$

where $l_2 = 1$. Plug in the expression of $\mathbf{A}_2$, the left hand of inequality (2.56) can be written as

$$\begin{aligned}&\mathbf{A}_2 + \mu\mathbf{I} - \frac{1}{1-\mu}\mathbf{b}_2\mathbf{b}_2^\top \\ &= \|\mathbf{n}\|^2(\mathbf{I} - \mathbf{X}\mathbf{X}^\dagger) + \mu\mathbf{I} - \frac{\mu}{1-\mu}\mathbf{n}\mathbf{n}^\top.\end{aligned}$$

Since $\mathbf{I} - \mathbf{X}\mathbf{X}^\dagger$ is a projection matrix, it is positive semi-definite. So, we only need to prove

$$\mu\mathbf{I} - \frac{\mu}{1-\mu}\mathbf{n}\mathbf{n}^\top \succ \mathbf{0}. \tag{2.57}$$

Since $\mathbf{n}\mathbf{n}^\top$ is rank-one and its non-zero eigenvalue is $\|\mathbf{n}\|^2$, it equals to proving $\|\mathbf{n}\|^2/(1-\mu) < 1$. To guarantee this inequality, we only need to make sure $\mu < 1 - \|\mathbf{n}\|^2$. Since $\|\mathbf{X}^\dagger\| \leq 1/\sigma_m$ and $\|\mathbf{d}\| \leq \eta$, we get $\|\mathbf{n}\|^2 = \|(\mathbf{X}^\dagger)^\top\mathbf{d}\|^2 \leq \|\mathbf{X}^\dagger\|^2\|\mathbf{d}\|^2 \leq \eta^2/\sigma_m^2 < 1$. By choosing $0 < \mu < 1 - \|\mathbf{n}\|^2 < 1$, we can ensure (2.55) and (2.56) are both satisfied, and hence inequality (2.53) is satisfied.

From Theorem 2.1, we know the optimal value of (2.51) is equivalent to the optimal value of problem (2.54). Problem (2.54) is a semidefinite programming problem, which is convex and can be easily solved by modern tools such as [90] and [91]. We now discuss how to find the optimal $\mathbf{c}$ which achieves this value. Suppose the optimal solution of problem (2.54) is (α^*, ν^*). Since, $h_2(\mathbf{c}) > 0$, we have $h_1(\mathbf{c}) \geq \alpha^* h_2(\mathbf{c})$ for any feasible $\mathbf{c}$. Hence, we can

compute the optimal solution of problem (2.51) by solving

$$\underset{\mathbf{c}}{\text{argmin}}: \quad h_1(\mathbf{c}) - \alpha^* h_2(\mathbf{c}) \tag{2.58}$$

$$\text{s.t.} \quad \|\mathbf{c}\|^2 \leq 1 \tag{2.59}$$

This problem is just a trust region problem. There are several existing methods to solve it efficiently. In this chapter, we employ the method described in [92].

Now, we turn to solve problem (2.48). Since (2.48) and (2.47) have similar structure, we can employ the methods described in Theorem 2.1 and (2.58) to find its optimal value and optimal solution for problem (2.48).

Until now, we have fully described how to solve the intermediate problems in the alternating optimization method. The following theorem shows that the proposed alternating optimization algorithm will converge. Suppose the generated sequence of solution is $\{\mathbf{c}^k, \mathbf{d}^k\}$, $k = 0, 1, \cdots$, and we have the following corollary:

Corollary 1. *The sequence* $\{\mathbf{c}^k, \mathbf{d}^k\}$ *admits a limit point* $\{\bar{\mathbf{c}}, \bar{\mathbf{d}}\}$ *and we have*

$$\lim_{k\to\infty} h(\mathbf{c}^k, \mathbf{d}^k) = h(\bar{\mathbf{c}}, \bar{\mathbf{d}}). \tag{2.60}$$

Furthermore, every limit point is a critical point, which means

$$\nabla h(\bar{\mathbf{c}}, \bar{\mathbf{d}})^\top \begin{bmatrix} \mathbf{c} - \bar{\mathbf{c}} \\ \mathbf{d} - \bar{\mathbf{d}} \end{bmatrix} \geq 0, \tag{2.61}$$

for any $\|\mathbf{c}\| \leq 1$ *and* $\|\mathbf{d}\| \leq \eta$.

Proof. We first give the proof of (2.60). Since the sequence $\{\mathbf{c}^k, \mathbf{d}^k\}$ lies in the compact set, $\{(\mathbf{c}, \mathbf{d}) \,|\, \|\mathbf{c}\| \leq 1, \|\mathbf{d}\| \leq \eta\}$, and according to the Bolzano-Weierstrass Theorem [93], $\{\mathbf{c}^k, \mathbf{d}^k\}$ must have limit points. Hence, there is a subsequence of $\{h^k\}$ which converges to $h(\bar{\mathbf{c}}, \bar{\mathbf{d}})$. As the objective is a continuous function with respect to $\mathbf{c}$ and $\mathbf{d}$, the compactness

of the constraint also implies the sequence of the objective value, $\{h^k\}$, is bounded from below. In addition, $\{h^k\}$ is a non-increasing sequence, which indicates that the sequence of the function value must converge. In summary, the sequence $\{h^k\}$ must converge to $h(\bar{\mathbf{c}}, \bar{\mathbf{d}})$. For the rest of the proof, please refer to Corollary 2 of [94] for more details. □

2.4 Numerical Examples

In this section, we test our adversarial attack strategies on practical regression problems. In the first regression task, we use seven international indexes to predict the returns of the Istanbul Stock Exchange [95]. The data set contains 536 data samples, which are the records of the returns of Istanbul Stock Exchange with seven other international indexes starting from Jun. 5, 2009 to Feb. 22, 2011. Also, we demonstrate how our attack impacts the quality of a regression task using the wine dataset [96].

2.4.1 Attacking One Specific Regression Coefficient

In this experiment, we attack the fourth regression coefficient of the Istanbul Stock Exchange dataset and try to make its magnitude large by solving problem (2.4). We use two strategies to attack this coefficient with a fixed energy budget $\eta = 0.2$. The first strategy is the one proposed in this chapter. As a comparison, we also use a random strategy to approximate the exhaustive search algorithm. In the random strategy, we randomly generate the adversarial data point with each entry being i.i.d. generated from a standard normal distribution. Then, we normalize its energy to be η. We repeat this random attack 10000 times and select the one with the smallest objective value. Hence, the random strategy is an approximation of the exhaustive search algorithm.

Fig. 2.1 shows the regression coefficients before and after our attack. The x-axis denotes the index of the regression coefficients and the y-axis indicates the value of the regression coefficients. In this figure, the 'orig' denotes the original regression coefficient, 'opt' represents

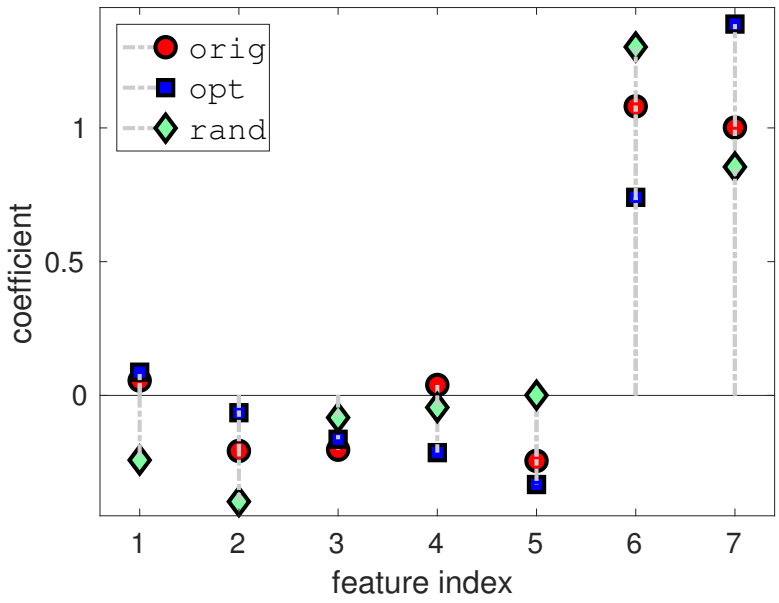


Figure 2.1: The regression coefficients before and after attacking the fourth regression coefficient with objective (2.5).

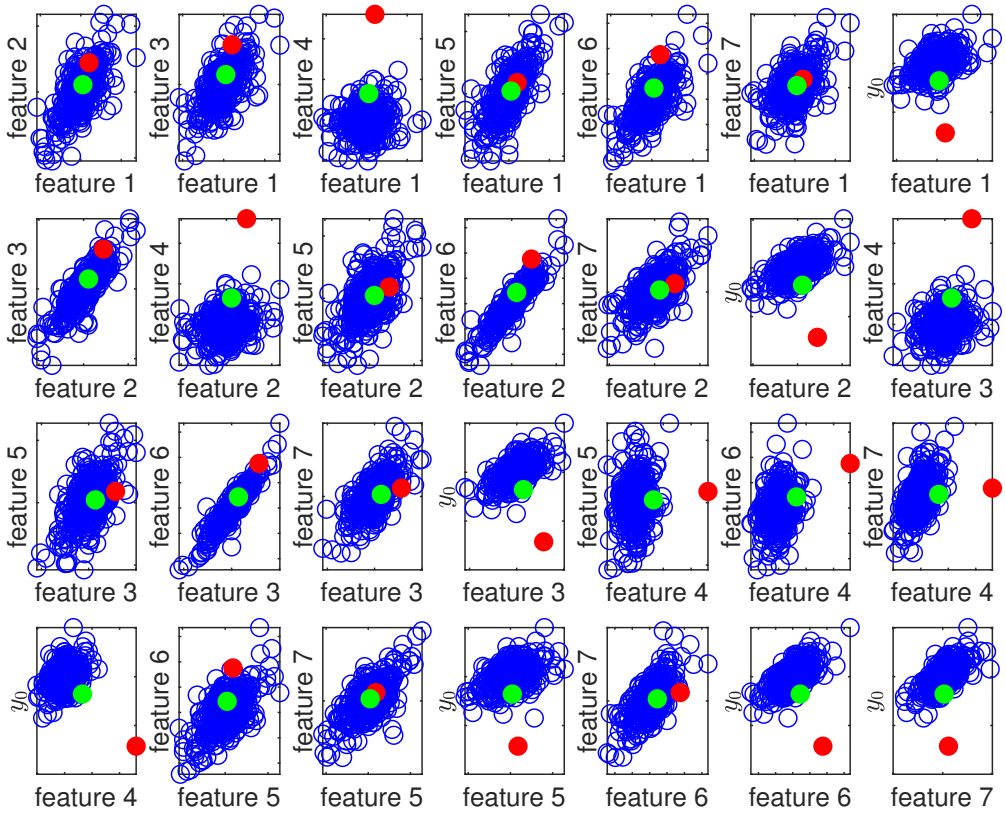


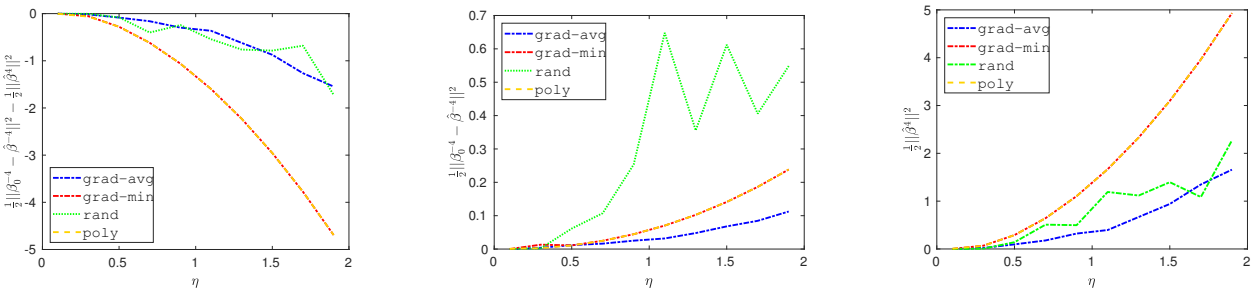
Figure 2.2: The scatter plot of the original data, the designed poisoning data, and the poisoning data after the repeating strategy.

the regression coefficient after attacking by our proposed optimal attack strategy, and 'rand' indicates the regression coefficient after attacking by the random attack strategy. From the figure we can see that our proposed adversarial attack strategy is much more efficient than the random attack strategy. One can also observe that by only adding one adversarial example, designed by the approach characterized in this chapter, one can dramatically change the value of a regression coefficient and hence change the importance of that explanatory variable.

Fig. 2.2 shows the original data points (in blue), the optimal adversarial data point (in red), and the adversarial data points after the 16 times repeating strategy (in green) in this experiment. In this figure, the x-axis and y-axis are two features that are specified by their corresponding axes labels (including the response value). The blue circle represents the original data, the solid red dot denotes the data point designed by our proposed method in Algorithm 1, and the solid green circle indicates our proposed poisoning data after 16 times of repeating. The figure demonstrates that the proposed adversarial data point may behave as an outlier. However, after our simple repeating strategy, the adversarial data points act just like normal data points. Hence, our repeating strategy can mitigate the adversarial data point being detected by the regression system.

2.4.2 Attacking without Changing Untargeted Regression Coefficients

From the numerical examples in the previous subsection, we can see the untargeted regression coefficients may change greatly while attacking one specific regression coefficient with an adversarial data point. For example, as demonstrated in Fig. 2.1, the sixth and seventh regression coefficients change significantly when we attack the fourth regression coefficient. To mitigate the undesirable changes of untargeted regression coefficients, we need more sophisticated attacking strategies. In this subsection, we will test different strategies with a more general objective function as demonstrated in Chapter 2.2.3. We also use the same



(a) (b) (c)

Figure 2.3: Attack the fourth regression coefficient with objective (2.30) and $\lambda = -1$ under different energy budgets.

data set as described in the previous subsection. We first try to attack the fourth regression coefficient to increase its importance while making only small changes to the rest of the regression coefficients. To accomplish this task, we aim to solve problem (2.30) with $\lambda = -1$. Given the energy budget, firstly, we use our semidefinite relaxation based algorithm to solve problem (2.37), and then follow Algorithm 2 to find the adversarial data point. For comparison, we also carry out the random attack strategy, in which we randomly generate the data point with each entry being i.i.d. according to the standard normal distribution. Then, we normalize its energy being η and added it to the original data points. We repeat these random attacks 10000 times and select the one with the smallest objective value. The third strategy is the projected gradient descent based strategy, where we use the projected gradient descent algorithm to solve (2.37) and follow similar steps of Algorithm 2 to find the adversarial data point. Projected gradient descent works much like the gradient descent except with an additional operation that projects the result of each step onto the feasible set after moving in the direction of negative gradient [97]. In our experiment, we use diminishing step-size, $1/(t+1)$. Since the projected gradient descent algorithm depends on the initial points heavily, given the energy budget, we repeat it 100 times with different random initial points and treat the average of its objective values as the objective value of this algorithm. Also, among the 100 times attacks, we record the one with the smallest objective value.

Fig. 2.3 shows the objective values under different energy budgets with different attacking

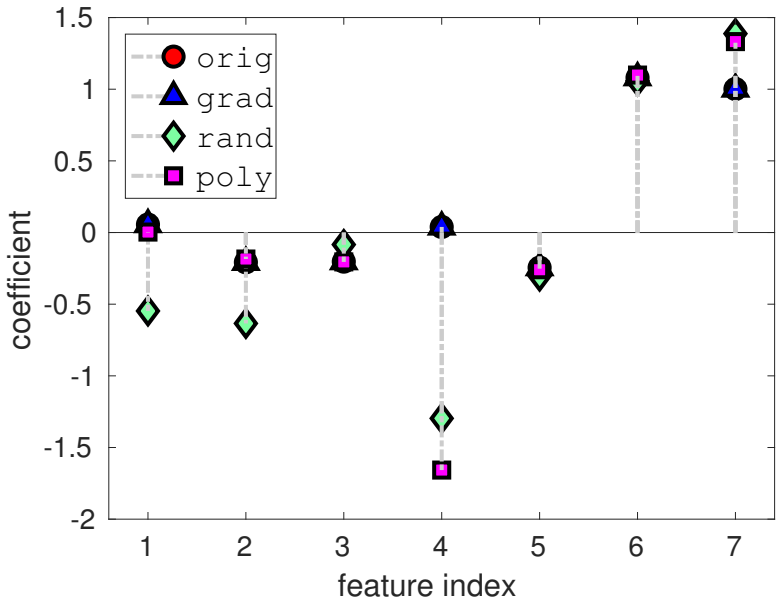
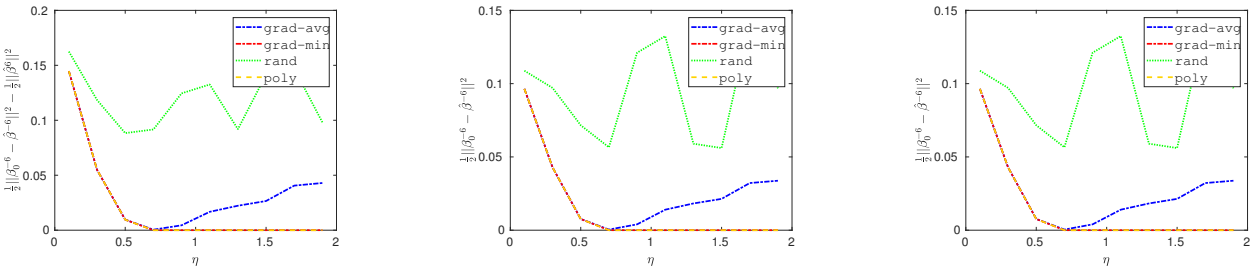


Figure 2.4: The regression coefficients before and after different kinds of strategies that attack the fourth regression coefficient with energy budget $\eta = 1$.

strategies and Fig. 2.4 demonstrates the regression coefficients after one of the attacks of different strategies with $\eta = 1$. In these figures, 'orig' is the original regression coefficient, 'rand' means the random strategy, 'poly' indicates our semidefinite relaxation strategy, 'grad-avg' is the average objective value of the 100 times attacks based on the projected gradient descent algorithm, and 'grad-min' is the one with the smallest objective value among the 100 times attacks based on the projected gradient descent algorithm. From these two figures, we can see our semidefinite relaxation based strategy performs much better than the other two strategies. Among the 100 times attacks based on the projected gradient descent, the minimal one can achieve similar objective values as our proposed attacks based on the semidefinite relaxation. In addition, in our experiment, our semidefinite relaxation method with relaxation order 2 or 3 can always lead to globally optimal solutions. Hence, the computational complexity of this method is still low. Fig. 2.4 also shows our relaxation based method leads to the largest magnitude of the fourth regression coefficient while keeping other regression coefficients almost unchanged.

In the second experiment, we attack the sixth regression coefficient and attempt to make



(a) (b) (c)

Figure 2.5: Attack the sixth regression coefficient with objective (2.30) and $\lambda = 1$ under different energy budgets.

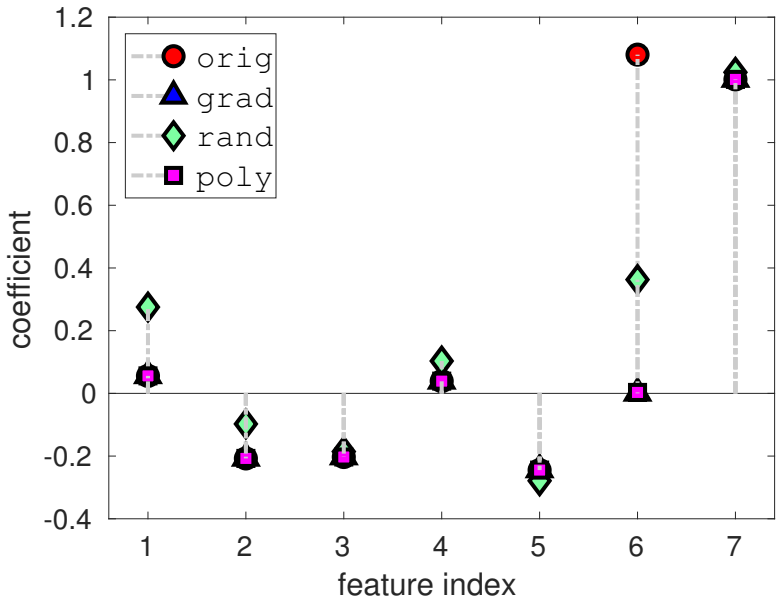


Figure 2.6: The regression coefficients after different kinds of strategies that attack the sixth regression coefficient with energy budget $\eta = 1$.

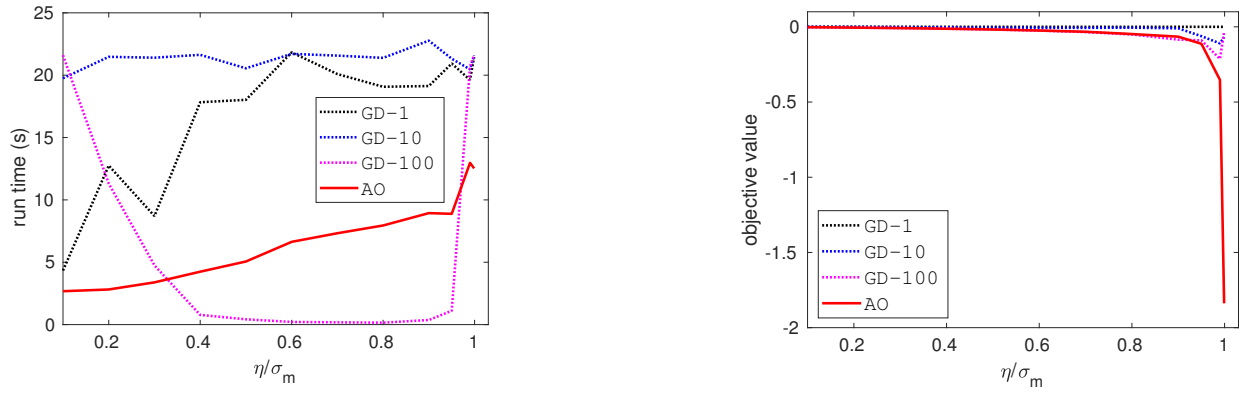
its magnitude small while keeping the change of the rest of the coefficients to be small. So, we set $\lambda = 1$ in problem (2.30) to achieve this goal. The settings of each strategy are similar to the ones in the first experiment. Fig. 2.5 shows the objective values with different strategies under different energy budgets and Fig. 2.6 demonstrates the regression coefficients after one of the attacks of those strategies respectively with energy budget $\eta = 1$. From Fig. 2.5 we know the projected gradient descent based strategy and the semidefinite relaxation based strategy achieve much lower objective values compared to the random attack strategy. Specifically, when the energy budget is smaller than 0.7, both of the two strategies

behave similarly. However, when the energy budget is larger than 0.7, the projected gradient descent based strategy leads to larger objective values as the energy budget grows. This is because the projected gradient descent algorithm tends to find solutions at the boundary of the feasible set. Only some attacks with good initialization can lead to the global minimum. By contrast, our semidefinite relaxation based strategy can find the globally optimal solutions with relaxation order 2 or 3. Thus, it gives the best performance among the three strategies. Fig. 2.6 also demonstrates our relaxation based method achieves the global optimum when $\eta = 1$ as it leads the sixth regression coefficient to zero and other regression coefficients to be unchanged.

2.4.3 Rank-one Attack

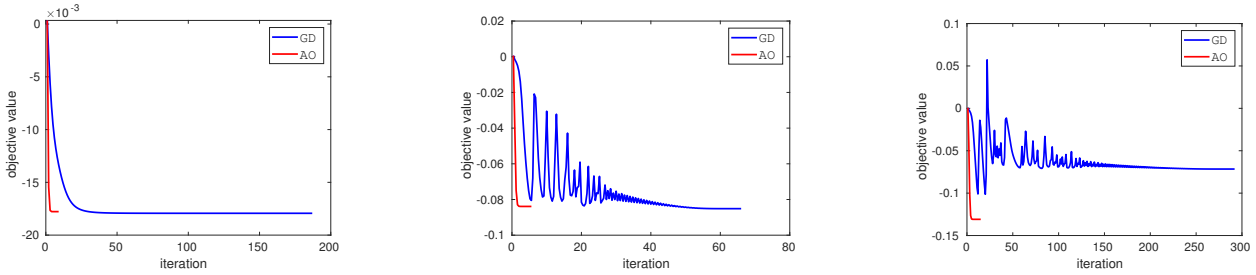
In this subsection, we carry out different rank-one attack strategies. Our goal is to minimize the magnitude of the fourth regression coefficient with objective (2.41). We compare two strategies: the projected gradient descent based strategy discussed in Chapter 2.4.2 and our proposed alternating optimization based strategy. For the projected gradient descent based strategies, we use different step sizes: $1/(1+t)$, $10/(1+t)$, and $100/(1+t)$. As our analysis shows, when the energy budget is larger than the smallest singular value, our objective can be minus infinity. Hence, in our experiment, we vary the energy budget from 0 to the smallest singular value, which is 0.053. Given a certain energy budget, we set all the algorithms with the same randomly initialized point and run these algorithms until they stop with the same convergence condition: two consecutive function values change too small or it reaches the maximal allowable iterations. We repeat this process 100 times and record their average objective values.

Fig. 2.7 (a) shows the averaged run times and Fig. 2.7 (b) illustrates objective values of the four algorithms, where 'GD-1', 'GD-10' and 'GD-100' stand for the projected gradient descent with stepsizes $1/(1+t)$, $10/(1+t)$, and $100/(1+t)$, respectively, and 'AO' denotes the proposed alternating optimization method. We carry out this experiment on a PC with four



(a) (b)

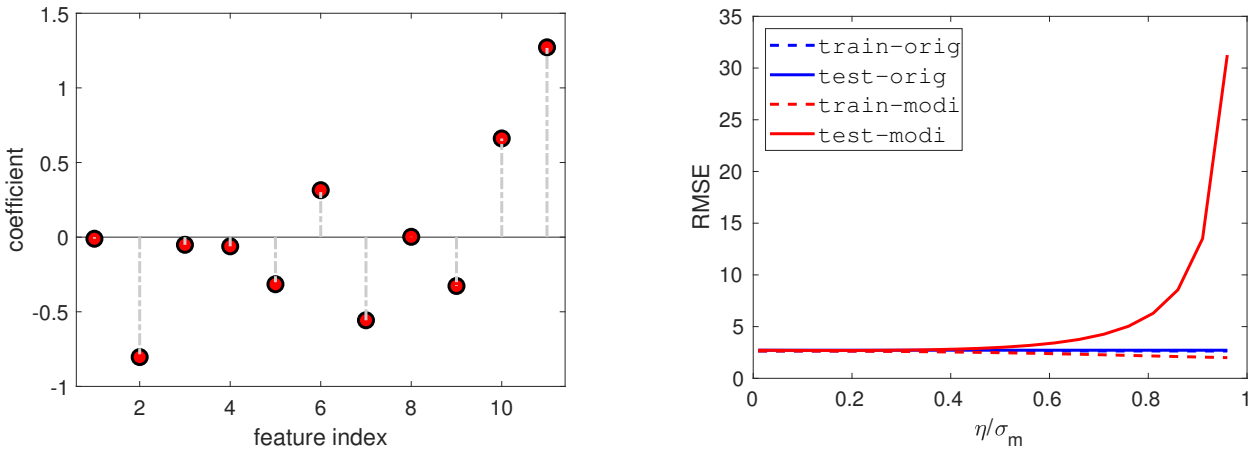
Figure 2.7: The averaged run times (Subfigure (a)) and the objective values (Subfigure (b)) of the projected gradient descent and the proposed alternating optimization method with different stepsizes.



(a) (b) (c)

Figure 2.8: The evolution of function values as the iteration increases with one typical run of projected gradient descent and alternating optimization algorithm.

Intel E3 CPUs. All the four algorithms have the same convergence condition: the absolute value of the difference of two consecutive objective values is less than 10^{-5}. Fig. 2.7 (a) shows that, as the energy budget increases, the run times of the alternating optimization, GD-1, and GD-10 increase. However, as the energy budget increases, the run times of GD-100 first decrease and then increase. This is due to the fact that a larger stepsize will result in a faster convergence rate while it may cause oscillation. Fig. 2.7 (b) shows that when the energy budget increases, the objectives decrease for both of these algorithms. Furthermore, the proposed alternating optimization based algorithm provides much smaller objective values, especially when the energy budget approaches the smallest singular value. When the energy budget approaches the smallest singular value, the gradient descent based



(a) (b)

Figure 2.9: The regression coefficient of the original data set (subfig (a)) and the RMSE on the training and test data set with different energy budgets (subfig (b)).

algorithm becomes very unstable. This is due to the fact that when the energy budget is large, the objective is very sensitive to the energy budget. So, a small stepsize may result in significant objective value change. This phenomena can be observed in Fig. 2.8, where it depicts the evolution of the objective values of 'AO' and 'GD-100' with the energy budget being $\eta/\sigma_m = 0.5$ (subfigure (a)), $\eta/\sigma_m = 0.9$ (subfigure (b)) and $\eta/\sigma_m = 0.95$ (subfigure (c)), respectively, and σ_m is the smallest singular value of the original feature matrix. From this figure we can see the alternating optimization based algorithm converges very fast while the projected gradient descent based algorithm becomes unstable when the energy budget is large. This is due to the fact that the objective of our alternating optimization based algorithm is guaranteed to be monotonically decreasing.

In the second experiment, we test our rank-one attack strategy on the wine dataset[96], which includes 11 chemical analysis of the red wine and its corresponding quality (ranging from 3 to 8). In this dataset, we have 1599 data samples and we randomly choose 80 percent of the data as the training set and the rest as the test data. We use linear regression to learn the regression coefficients on the training data and then use these regression coefficients on the test data to predict the quality of the test data. We use the root mean square error (RMSE) to measure the goodness of predicting both on the training data and test data.

We use the rank-one attack strategy proposed in this chapter on the training data with the target of maximizing the eighth regression coefficient (corresponding to the density feature). We carry out the attack with different energy budgets ranging from 0 to the smallest singular value of the feature matrix of training data.

Fig. 2.9 (a) illustrates the original regression coefficients without attack. The magnitude of the eighth regression coefficient is very small. It reveals that the eighth feature is not important compared to other features. Fig. 2.9 (b) shows the RMSE on the training data and test data using different energy budget with and without attacking the eighth regression coefficient. 'train-orig' and 'test-orig' represent the RMSE on the training and test data without attacking the training data. 'train-modi' and 'test-modi' denote the RMSE on the training and test data when we conduct our rank-one attacking on the training dataset. This figure demonstrates that, even though the RMSE on the attacked training data is low, the model based on the attacked features performs extremely badly on the test data. It illustrates that attacking the regression coefficient not only misleads the interpretation of the model but also has significant impact on the performance of the model.

2.5 Summary

In this chapter, we have investigated the adversarial robustness of linear regression problems. Particularly, we have given the closed-form solution when we attack one specific regression coefficient with a limited energy budget. Furthermore, we have considered a more complex objective where we attack one of the regression coefficients while trying to keep the rest of the regression coefficients to be unchanged. We have formulated this problem as a multivariate polynomial optimization problem and introduced the semidefinite relaxation method to solve it. Finally, we have studied a more powerful adversary who can make a rank-one modification on the feature matrix. To take the advantage of the rank-one structure, we have proposed an alternating optimization algorithm to solve this problem. The numerical

examples demonstrated that our proposed closed-form solution and the semidefinite relaxation based strategies can find the globally optimal solutions and the alternating optimization based strategy provides better solutions, faster convergence, and more stable behavior compared to the projected gradient descent based strategy. We should also note that the solutions are "optimal" under the specific objectives mentioned in the chapter. Clearly, if the goal of the attacker is changed, then the optimal attack strategy will be different.

Chapter 3

On the Adversarial Robustness of LASSO Based Feature Selection

3.1 Introduction

In this chapter, we investigate the adversarial robustness of LASSO based feature selection problem. We introduce a smooth approximation of the ℓ_1 norm and use the projected gradient descent to design the modifications on the feature matrix and response values in order to manipulate the regression coefficient. This chapter is organized as follows. In Chapter 3.2, we describe the precise problem formulation based on the ordinary LASSO feature selection method. In Chapter 3.3, we introduce our method to solve this problem. In Chapter 3.4, we extend our method to attack the group LASSO and the sparse group LASSO based feature selection methods. In Chapter 3.5, we provide comprehensive numerical experiments with both synthetic data and real data to illustrate the results obtained in this paper. Finally, we offer concluding remarks in Chapter 3.6.

3.2 Problem Formulation

In this section we provide the problem formulation of adversarial attack against the ordinary LASSO based feature selection.

Given the data set $\{(y_0^k, \mathbf{x}_0^k)\}_{k=1}^n$, where n is the number of data samples, y_0^k is the response value of data sample k, $\mathbf{x}_0^k \in \mathbb{R}^m$ denotes the feature vector of data sample k, and each element of $\mathbf{x}_0^k$ is called a feature of the data sample. Through the data samples, we attempt to learn a sparse representation of the response values from the features. The LASSO algorithm learns a sparse regression coefficient, $\boldsymbol{\beta}_0$, by solving

$$\boldsymbol{\beta}_0 = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \|\mathbf{y}_0 - \mathbf{X}_0\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1, \tag{3.1}$$

where the response vector $\mathbf{y}_0 = [y_0^1, y_0^2, \ldots, y_0^n]^\top$, the feature matrix $\mathbf{X}_0 = [\mathbf{x}_0^1, \mathbf{x}_0^2, \ldots, \mathbf{x}_0^n]^\top$, $\|\cdot\|_1$ denotes the ℓ_1 norm, and λ is the trade-off parameter to determine the relative goodness of fitting and sparsity of $\boldsymbol{\beta}_0$ [40]. The locations of the non-zero elements of the sparse regression coefficients indicate the corresponding selected features.

In this chapter, we assume that there is an adversary who is trying to manipulate the learned regression coefficients and thus maneuver the selected features by carefully modifying the response values or the feature matrix. We denote the modified response value vector as $\mathbf{y}$ and denote the modified feature matrix as $\mathbf{X}$. Further, we assume that the adversary's modification is constrained by the ℓ_p norm ($p \geq 1$). This means we have $\|\mathbf{y} - \mathbf{y}_0\|_p \leq \eta_y$, and $\|\mathbf{X} - \mathbf{X}_0\|_p \leq \eta_x$, where η_y is the energy budget for the modification of the response values, and η_x is the energy budget for the modification of the feature matrix. For a vector, $\|\cdot\|_p$ denotes the ℓ_p norm of the vector; for a matrix, $\|\cdot\|_p$ denotes the ℓ_p norm of the vectorization of the matrix. As a result, the manipulated regression coefficients, $\hat{\boldsymbol{\beta}}$, are learned from the modified data set $(\mathbf{y}, \mathbf{X})$ by solving the following LASSO problem

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1. \tag{3.2}$$

The goal of the adversary is to suppress or promote some of the regression coefficients while keeping the change of the remaining coefficients to be minimum. If it wants to suppress the ith regression coefficient, we minimize $s_i \cdot \hat{\beta}_i^2$, where $s_i > 0$ is the predefined weight parameter. If it aims to promote the ith regression coefficient, we minimize $e_i \cdot \hat{\beta}_i^2$, where $e_i < 0$ is the weight parameter. To make the changes to the ith regression coefficient as small as possible, we minimize $\mu_i \cdot (\hat{\beta}_i - \beta_0^i)^2$, where $\mu_i > 0$ is a user defined parameter to measure how much effort we put on keeping the ith regression coefficients intact. Moreover, we denote the set of indices of coefficients which are suppressed, promoted, and not changed as S, E, and U, respectively. In summary, the objective of the adversary is:

$$\min_{\hat{\boldsymbol{\beta}}} \frac{1}{2}(\hat{\boldsymbol{\beta}} - \boldsymbol{\nu})^\top \mathbf{H}(\hat{\boldsymbol{\beta}} - \boldsymbol{\nu}), \tag{3.3}$$

where $\nu_i = \beta_0^i$ if $i \in U$, otherwise $\nu_i = 0$, $\mathbf{H} = \text{diag}(\mathbf{h})$, $\text{diag}(\mathbf{h})$ is the diagonal matrix with its diagonal elements being $\mathbf{h}$, and $h_i = \mu_i$ for $i \in U$, $h_i = s_i$ for $i \in S$ and $h_i = e_i$ for $i \in E$.

Considering the energy constraints of the adversary and the fact that $\hat{\boldsymbol{\beta}}$ is a function of $\mathbf{y}$ and $\mathbf{X}$, we need to solve the following bi-level optimization problem to obtain the optimal attack strategy.

$$\min_{\mathbf{y} \in \mathcal{C}_y, \mathbf{X} \in \mathcal{C}_x} \quad f(\mathbf{y}, \mathbf{X}) \tag{3.4}$$

$$\text{s.t.} \quad \hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\text{argmin}} \, \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1, \tag{3.5}$$

where

$$\mathcal{C}_y = \{\mathbf{y} \,|\, \|\mathbf{y} - \mathbf{y}_0\|_p \leq \eta_y\},$$

$$\mathcal{C}_x = \{\mathbf{X} \,|\, \|\mathbf{X} - \mathbf{X}_0\|_p \leq \eta_x\},$$

and $f(\mathbf{y}, \mathbf{X}) = \frac{1}{2}(\hat{\boldsymbol{\beta}} - \boldsymbol{\nu})^\top \mathbf{H}(\hat{\boldsymbol{\beta}} - \boldsymbol{\nu})$.

3.3 Algorithm

In this section, we investigate problem (3.4) and present our projected gradient descent method to solve this problem.

In problem (3.4), the objective is a function of $\hat{\boldsymbol{\beta}}$. However, the relationship between $(\mathbf{y}, \mathbf{X})$ and $\hat{\boldsymbol{\beta}}$ is determined by the lower-level optimization problem. This makes our objective a very complicated function of $(\mathbf{y}, \mathbf{X})$ and in general (3.4) is not convex. To illustrate this, we consider a simplified version of this problem in which we have scalar y and x. In this case, our problem can be written as

$$\begin{aligned} \min_{x \in \mathcal{C}_x,\, y \in \mathcal{C}_y} \quad & h\hat{\beta}, \\ \text{s.t.} \quad & \hat{\beta} = \operatorname*{argmin}_{\beta} \quad (y - x\beta)^2 + \lambda|\beta|. \end{aligned}$$

The solution to the lower-level optimization problem is $\hat{\beta} = \operatorname{sgn}(y/x)(y/x - \lambda/(2x^2))_+$, where $\operatorname{sgn}(\cdot)$ is the sign function and $(\cdot)_+$ takes the positive part of the argument. Hence, our problem can be simplified as

$$\min_{x \in \mathcal{C}_x,\, y \in \mathcal{C}_y} \quad h[(y/x - \lambda/(2x^2))_+]^2.$$

It is easy to verify that this problem is not convex. To solve this bi-level optimization problem, we need to first solve the lower-level optimization problem to determine the dependence between $(\mathbf{y}, \mathbf{X})$ and $\hat{\boldsymbol{\beta}}$. Then, we can use the gradient descent method to solve this bi-level optimization problem. Since the lower-level problem is convex [40], it can be represented by its first order optimality condition. The corresponding first order optimality condition with respect to the lower-level optimization problem is:

$$\mathbf{0} \in 2\mathbf{X}^\top(\mathbf{X}\boldsymbol{\beta} - \mathbf{y}) + \lambda \partial \|\boldsymbol{\beta}\|_1, \tag{3.6}$$

when $\boldsymbol{\beta} = \hat{\boldsymbol{\beta}}$, where $\partial\|\cdot\|_1$ is the subgradient of the ℓ_1 norm. We denote the right hand of (3.6) as $q(\boldsymbol{\beta}, \mathbf{y}, \mathbf{X})$.

If $q(\boldsymbol{\beta}, \mathbf{y}, \mathbf{X})$ is a continuously differentiable function and its Jacobian matrix with respect to $\boldsymbol{\beta}$ is invertible, the first order condition defines a one-to-one mapping from $(\mathbf{y}, \mathbf{X})$ to $\boldsymbol{\beta}$, and by the implicit function theorem [98], we can calculate the gradient of $\boldsymbol{\beta}$ with respect to $\mathbf{y}$ and $\mathbf{X}$. Unfortunately, in our case, $q(\boldsymbol{\beta}, \mathbf{y}, \mathbf{X})$ is not differentiable at the point with $\beta_i = 0$. Moreover, (3.5) does not always determine a single valued mapping from $(\mathbf{y}, \mathbf{X})$ to $\boldsymbol{\beta}$. For example, when $\lambda \geq \|\mathbf{X}^\top \mathbf{y}\|_\infty$, we always have $\boldsymbol{\beta} = \mathbf{0}$.

To circumvent these difficulties, we transform the lower-level optimization problem to the following equivalent linear inequality constrained quadratic programming [99]:

$$\underset{\boldsymbol{\beta}, \mathbf{u}}{\operatorname{argmin}} \quad \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \sum_{i=1}^{m} u_i \tag{3.7}$$

$$\text{s.t.} \quad -u_i \leq \beta_i \leq u_i,\ i = 1, 2, \ldots, m, \tag{3.8}$$

where $\mathbf{u} = [u_1, u_2, \ldots, u_m]^\top$. Following [99], we can apply the interior-point method to solve (3.7). In particular, we solve the penalized problem:

$$\underset{\boldsymbol{\beta}, \mathbf{u}}{\operatorname{argmin}} \quad \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \sum_{i=1}^{m} u_i + \frac{1}{t}\Phi(\boldsymbol{\beta}, \mathbf{u}), \tag{3.9}$$

where $\Phi(\boldsymbol{\beta}, \mathbf{u}) = -\sum_{i=1}^{m} \log(u_i^2 - \beta_i^2)$ is the penalty function for the constraints of (3.7) and t is the penalty parameter. Solution of problem (3.9) converges to (3.2) if we follow the central path as t varies from 0 to ∞, where the central path is defined as the set of solution to (3.9) for different $t > 0$ [25].

Instead of using the first order optimality condition of (3.6), we utilize the first order

optimality condition of (3.9), which are

$$2\mathbf{X}^\top(\mathbf{X}\boldsymbol{\beta}-\mathbf{y})+\frac{1}{t}\nabla_{\boldsymbol{\beta}}\Phi=\mathbf{0}, \tag{3.10}$$

$$\lambda\mathbf{1}-\frac{1}{t}\nabla_{\mathbf{u}}\Phi=\mathbf{0}, \tag{3.11}$$

where

$$\nabla_{\boldsymbol{\beta}}\Phi=\begin{bmatrix}2\beta_1/(u_1^2-\beta_1^2),\\ \vdots\\ 2\beta_m/(u_m^2-\beta_m^2)\end{bmatrix}$$

and

$$\nabla_{\mathbf{u}}\Phi=\begin{bmatrix}2u_1/(u_1^2-\beta_1^2)\\ \vdots\\ 2u_m/(u_m^2-\beta_m^2)\end{bmatrix}.$$

Let us denote the first order optimality condition as $\mathbf{g}(\mathbf{y},\mathbf{X},\boldsymbol{\beta},\mathbf{u})=\mathbf{0}$. According to the implicit function theorem, the derivative of $\boldsymbol{\beta}$ with respect to $\mathbf{y}$ can be computed as

$$\nabla_{\mathbf{y}}\boldsymbol{\beta}=-[\mathbf{J}^{-1}]_{1:m}\nabla_{\mathbf{y}}\mathbf{g}, \tag{3.12}$$

where $[\mathbf{J}^{-1}]_{1:m}$ denotes the first m rows of $\mathbf{J}_{1:m}^{-1}$, $\mathbf{J}=[\nabla_{\boldsymbol{\beta}}\mathbf{g},\nabla_{\mathbf{u}}\mathbf{g}]$ is the Jacobian matrix of $\mathbf{g}(\mathbf{y},\mathbf{X},\boldsymbol{\beta},\mathbf{u})$ with respect to $\boldsymbol{\beta}$ and $\mathbf{u}$,

$$\nabla_{\mathbf{y}}\mathbf{g}=\begin{bmatrix}-2\mathbf{X}^\top\\ \mathbf{0}\end{bmatrix}, \tag{3.13}$$

$$\nabla_{\boldsymbol{\beta}}\mathbf{g}=\begin{bmatrix}2\mathbf{X}^\top\mathbf{X}+\mathbf{D}_1\\ \mathbf{D}_2\end{bmatrix}, \tag{3.14}$$

$$\nabla_{\mathbf{u}}\mathbf{g}=\begin{bmatrix}\mathbf{D}_2\\ \mathbf{D}_1\end{bmatrix}, \tag{3.15}$$

with

$$\mathbf{D}_1 = \frac{1}{t}\mathrm{diag}\big(2(u_1^2+\beta_1^2)/(u_1^2-\beta_1^2)^2, \ldots, 2(u_m^2+\beta_m^2)/(u_m^2-\beta_m^2)^2\big),$$
$$\mathbf{D}_2 = \frac{1}{t}\mathrm{diag}\big(-4u_1\beta_1/(u_1^2-\beta_1^2)^2, \ldots, -4u_m\beta_m/(u_m^2-\beta_m^2)^2\big).$$

Also, according to (3.10), (3.11), and the implicit function theorem, the derivative of $\boldsymbol{\beta}$ with respect to $\mathbf{X}$ can be calculated as

$$\nabla_{\mathbf{X}}\boldsymbol{\beta} = -[\mathbf{J}^{-1}]_{1:m}\nabla_{\mathbf{X}}\mathbf{g}, \tag{3.16}$$

where $\nabla_{\mathbf{X}}\mathbf{g} \in \mathbb{R}^{2m\times(mn)}$ with

$$\frac{\partial g_i}{\partial X_{kl}} = \begin{cases} 2\delta_{li}(\mathbf{X}\boldsymbol{\beta}-\mathbf{y})_k + 2X_{ki}\beta_l, & \text{if } i \le m \\ 0, & \text{if } i > m \end{cases} \tag{3.17}$$

with δ_{li} being the Kronecker delta function

$$\delta_{li} = \begin{cases} 1, & \text{if } i = l, \\ 0, & \text{if } i \neq l, \end{cases}$$

and $(\mathbf{X}\boldsymbol{\beta}-\mathbf{y})_k$ being the kth element of the vector $(\mathbf{X}\boldsymbol{\beta}-\mathbf{y})$.

To calculate the gradient of $\boldsymbol{\beta}$ with respect to $\mathbf{y}$ and $\mathbf{X}$, we first need to find the inverse of the Jacobian matrix. The Jacobian matrix is a 2×2 block matrix,

$$\mathbf{J} = \begin{bmatrix} 2\mathbf{X}^\top\mathbf{X}+\mathbf{D}_1 & \mathbf{D}_2 \\ \mathbf{D}_2 & \mathbf{D}_1 \end{bmatrix}.$$

This block structure makes the inverse of $\mathbf{J}$ admit a simple form [100]:

$$\mathbf{J}^{-1} = \begin{bmatrix} \tilde{\mathbf{J}}_{11} & \tilde{\mathbf{J}}_{12} \\ \tilde{\mathbf{J}}_{21} & \tilde{\mathbf{J}}_{22} \end{bmatrix}, \tag{3.18}$$

where $\tilde{\mathbf{J}}_{11} = (2\mathbf{X}^\top\mathbf{X} + 2\mathbf{D})^{-1}$ with $\mathbf{D} = 1/t \cdot \mathrm{diag}\big(1/(u_1^2+\beta_1^2), \ldots, 1/(u_m^2+\beta_m^2)\big)$, $\tilde{\mathbf{J}}_{12} = -\tilde{\mathbf{J}}_{11}\mathbf{D}_2\mathbf{D}_1^{-1}$, $\tilde{\mathbf{J}}_{21} = -\mathbf{D}_1^{-1}\mathbf{D}_2\tilde{\mathbf{J}}_{11}$, and $\tilde{\mathbf{J}}_{22} = \mathbf{D}_1^{-1} + \mathbf{D}_1^{-1}\mathbf{D}_2\tilde{\mathbf{J}}_{11}\mathbf{D}_2\mathbf{D}_1^{-1}$. With this explicit expression of the Jacobian matrix and note that the elements from $m+1$ to $2m$ are zero both for $\nabla_{\mathbf{y}}\mathbf{g}$ and $\nabla_{\mathbf{X}}\mathbf{g}$, we have

$$\nabla_{\mathbf{y}}\boldsymbol{\beta} = \big(\mathbf{X}^\top\mathbf{X} + \mathbf{D}\big)^{-1}\mathbf{X}^\top, \tag{3.19}$$

and

$$\frac{\partial\boldsymbol{\beta}}{\partial X_{kl}} = \left[\frac{\partial\beta_1}{\partial X_{kl}}, \frac{\partial\beta_2}{\partial X_{kl}}, \ldots, \frac{\partial\beta_m}{\partial X_{kl}}\right]^\top, \tag{3.20}$$

with

$$\frac{\partial\beta_i}{\partial X_{kl}} = \sum_j -(\mathbf{X}^\top\mathbf{X} + \mathbf{D})^{-1}_{ij}\frac{\partial g_j}{\partial X_{kl}}.$$

Using the chain rule, we have the gradient of f with respect to $\mathbf{y}$ and $\mathbf{X}$:

$$\nabla_{\mathbf{y}} f(\mathbf{y}, \mathbf{X}) = \nabla_{\mathbf{y}}\boldsymbol{\beta}^\top\mathbf{H}(\boldsymbol{\beta} - \boldsymbol{\nu})\Big|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}} \tag{3.21}$$

and

$$\frac{\partial f(\mathbf{y}, \mathbf{X})}{\partial X_{kl}} = (\boldsymbol{\beta} - \boldsymbol{\nu})^\top\mathbf{H}\frac{\partial\boldsymbol{\beta}}{\partial X_{kl}}\Big|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}}. \tag{3.22}$$

Now, we know the gradients of our objective function (3.4). With the help of this gradient information, we can use a variety of gradient based optimization methods. Since our problem is a constrained optimization problem, we resort to the projected gradient descent method.

Algorithm 4 The Projected Gradient Descent Algorithm

Input: data set $\{(y_0^i, \mathbf{x}_0^i)\}_{i=1}^n$, trade off parameter λ in (3.1), energy budget η_y, η_x, ℓ_p norm, and step-size parameter γ_k.
solve $\boldsymbol{\beta}_0$ via (3.1), set up feature sets S, E, U and their corresponding parameters $\mathbf{s}$, $\mathbf{e}$, $\boldsymbol{\mu}$; use those parameters to define the objective function $f(\mathbf{y}, \mathbf{X})$ in (3.4).
Initialize set the number of iterations $k = 0$ and randomly initialize $\mathbf{y}_k = \mathbf{y}_0$, $\mathbf{X}_k = \mathbf{X}_0$.
Do
solve $\hat{\boldsymbol{\beta}}$ according to (3.9),
compute the gradients: $\nabla_{\mathbf{y}} f(\mathbf{y}_k, \mathbf{X}_k)$ according to (3.21) and $\nabla_{\mathbf{X}} f(\mathbf{y}_k, \mathbf{X}_k)$ according to (3.22),
update:
$\mathbf{y}_{k+1} = \mathrm{Proj}_{\mathcal{C}_y}\big(\mathbf{y}_k - \gamma_k \nabla_{\mathbf{y}} f(\mathbf{y}_k, \mathbf{X}_k)\big)$,
update:
$\mathbf{X}_{k+1} = \mathrm{Proj}_{\mathcal{C}_x}\big(\mathbf{X}_k - \gamma_k \nabla_{\mathbf{X}} f(\mathbf{y}_k, \mathbf{X}_k)\big)$,
set $k = k + 1$,
While convergence conditions are not met.
Output: $\mathbf{y}_k, \mathbf{X}_k$.

We have summarized it in Algorithm 4. The main concept of the projected gradient descent algorithm is that we first take a gradient step, project it onto the feasible set, and then take an α_t step toward the projected point. In this algorithm, $\mathrm{Proj}_{\mathcal{C}_y}(\cdot)$ and $\mathrm{Proj}_{\mathcal{C}_x}(\cdot)$ represent the projection operators that project a point onto the feasible set $\mathcal{C}_y$ and $\mathcal{C}_x$, respectively. $\mathcal{C}_y$ and $\mathcal{C}_x$ are ℓ_p balls with radius η_y and η_x respectively. In the following, we will discuss the expressions of the projection onto three commonly used ℓ_p norm balls, where $p = 1, 2, \infty$, with unit radius and its center being the origin. We denote the projection onto the unit ℓ_p norm ball as $\mathrm{Proj}_{\mathbb{B}_{\ell_p}}(\cdot)$.

Case 1: Project onto the ℓ_1 unit norm ball. $\mathrm{Proj}_{\mathbb{B}_{\ell_1}}(\mathbf{x}) = \mathbf{z}^*$, where $\mathbf{z}^*$ is the solution to the following convex problem

$$\begin{aligned} \mathbf{z}^* = \underset{\mathbf{z}}{\operatorname{argmin}} \quad & \|\mathbf{z} - \mathbf{x}\|_2 \\ \text{s.t.} \quad & \|\mathbf{z}\|_1 \leq 1. \end{aligned}$$

Here $\mathbf{x}$ is the point to be projected. It can be efficiently solved via its dual with complexity $\mathcal{O}(m)$ [101].

Case 2: Project onto the ℓ_2 unit norm ball. In this case, we have a very simple closed-form solution

$$\text{Proj}_{\mathbb{B}_{\ell_2}}(\mathbf{x}) = \mathbf{x}/\max\{1, \|\mathbf{x}\|_2\}. \tag{3.23}$$

Case 3: Project onto the ℓ_∞ unit norm ball. In this case, we also have a very simple closed-form solution:

$$\text{Proj}_{\mathbb{B}_{\ell_\infty}}(\mathbf{x}) = \mathbf{z}^*, \tag{3.24}$$

where $\mathbf{z}^* = [z_1^*, \ldots, z_m^*]^\top$ and

$$z_i^* = \begin{cases} -1, & \text{if } x_i \leq -1, \\ x_i, & \text{if } |x_i| < 1, \\ 1, & \text{if } x_i \geq 1. \end{cases}$$

With these expressions of the projection, we can easily obtain the expressions of $\text{Proj}_{\mathcal{C}_y}(\cdot)$ and $\text{Proj}_{\mathcal{C}_x}(\cdot)$ by simply performing a geometric translation.

3.4 Adversarial Attacks against Group LASSO and Sparse Group LASSO

In this section, we will extend the method developed in Chapter 3.3 to design an optimal attack strategy towards two other popular LASSO based feature selection methods: group LASSO and sparse group LASSO. We also note that recent Bayesian based sparse learning methods obtain superior performance by incorporating the sparse and group sparse properties [102–104]. However, in this chapter, we will focus on the LASSO based methods.

3.4.1 Adversarial Attacks Against Group LASSO

Many of the sparse signals such as speech signal [105], frequency hopping spectrum [103], and functional brain network [106, 107], possess additional group structures. Specifically, these features are divided by groups and the features in the same group either contribute to the target simultaneously or not. To select the most useful features, it is better to exploit these additional structures [47]. The group LASSO imposes a group-wise sparsity structure, i.e., only a few groups have nonzero entries. This group-wise sparsity guides us to select better features, such as in splice site detection [108] and hyperspectral image classification [44]. The group-wise sparsity structure can be promoted by solving the following group LASSO problem:

$$\min_{\boldsymbol{\beta}} \quad \left\| \mathbf{y} - \sum_{l=1}^{L} \mathbf{X}_l \boldsymbol{\beta}_l \right\|_2^2 + \lambda \sum_{l=1}^{L} \sqrt{p_l} \|\boldsymbol{\beta}_l\|_2. \tag{3.25}$$

Here the feature matrix $\mathbf{X}$ is divided into L groups, each of which $\mathbf{X}_l \in \mathbb{R}^{n \times p_l}$, $\sum_{l=1}^{L} p_l = m$, and $\boldsymbol{\beta} = [\boldsymbol{\beta}_1^\top, \boldsymbol{\beta}_2^\top, \ldots, \boldsymbol{\beta}_L^\top]^\top$. The regularization term $\lambda \sum_{l=1}^{L} \sqrt{p_l} \|\boldsymbol{\beta}_l\|_2$ is used to promote the group-wise sparse structure, and λ is the penalty parameter to control the sparsity level and goodness of fitting.

Considering our attack target and the energy budget constraints for modifying the response values and the feature matrix, the design of optimal feature manipulation attacks for the group LASSO can be cast as a bi-level optimization:

$$\begin{aligned} \min_{\mathbf{y} \in \mathcal{C}_y, \mathbf{X} \in \mathcal{C}_x} \quad & \frac{1}{2} (\hat{\boldsymbol{\beta}} - \boldsymbol{\nu})^\top \mathbf{H} (\hat{\boldsymbol{\beta}} - \boldsymbol{\nu}) \\ \text{s.t.} \quad & \hat{\boldsymbol{\beta}} = \operatorname*{argmin}_{\boldsymbol{\beta}} \quad \left\| \mathbf{y} - \sum_{l=1}^{L} \mathbf{X}_l \boldsymbol{\beta}_l \right\|_2^2 + \lambda \sum_{l=1}^{L} \sqrt{p_l} \|\boldsymbol{\beta}_l\|_2, \end{aligned} \tag{3.26}$$

where $\boldsymbol{\nu}$ and $\mathbf{H}$ are defined the same as in problem (3.3).

To solve this bi-level optimization problem, we also first consider the lower-level group

LASSO problem. The group LASSO is a convex optimization problem, which is equivalent to the following quadratic programming with conic constraints:

$$\begin{aligned} \underset{\boldsymbol{\beta},\boldsymbol{\alpha}}{\text{argmin}} \quad & \left\| \mathbf{y} - \sum_{l=1}^{L} \mathbf{X}_l \boldsymbol{\beta}_l \right\|_2^2 + \sum_{l=1}^{L} \lambda_l \alpha_l \\ \text{s.t.} \quad & \|\boldsymbol{\beta}_l\|_2 \leq \alpha_l, \quad l = 1, 2, \ldots, L, \end{aligned} \tag{3.27}$$

where $\lambda_l = \lambda\sqrt{p_l}$ and $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_L]^\top$. To solve this problem, we can utilize the similar interior-point method we have employed for the ordinary LASSO problem in Chapter 3.3. In particular, we solve a series of the minimization problems: min f_t, as t gradually grows, where

$$f_t = \left\| \mathbf{y} - \sum_{l=1}^{L} \mathbf{X}_l \boldsymbol{\beta}_l \right\|_2^2 + \sum_{l=1}^{L} \lambda_l \alpha_l - 1/t \sum_{l=1}^{L} \log(\alpha_l^2 - \|\boldsymbol{\beta}_l\|_2^2).$$

Since this interior-point objective f_t is a convex function, the minimization problem is equal to its first order optimality condition:

$$\begin{aligned} \nabla_{\boldsymbol{\beta}_l} f_t &= \mathbf{X}_l^\top \left(\sum_{l=1}^{L} \mathbf{X}_l \boldsymbol{\beta}_l - \mathbf{y} \right) + \frac{1}{t} \frac{1}{\alpha_l^2 - \|\boldsymbol{\beta}_l\|_2^2} \boldsymbol{\beta}_l = \mathbf{0}, \\ \frac{\partial f_t}{\partial \alpha_l} &= \lambda_l - \frac{2}{t} \frac{\alpha_l}{\alpha_l^2 - \|\boldsymbol{\beta}_l\|_2^2} = 0, \qquad \text{for } l = 1, 2, \ldots, L. \end{aligned}$$

To derive the gradients of $\boldsymbol{\beta}$ with respect to $\mathbf{y}$ and $\mathbf{X}$, we can apply the implicit function theorem on the first order optimality condition. First, we need to compute the Jacobian matrix of the function on the left of the first order optimality condition. The derivative of

$\nabla_{\boldsymbol{\beta}} f_t$ with respect to $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$ can be computed by

$$\nabla_{\boldsymbol{\beta}_j} \nabla_{\boldsymbol{\beta}_i} f_t = \begin{cases} 2\mathbf{X}_i^\top \mathbf{X}_j, & \text{for } i \neq j, \\ 2\mathbf{X}_i^\top \mathbf{X}_j + \frac{1}{t} \frac{(\alpha_i^2 - \boldsymbol{\beta}_i^\top \boldsymbol{\beta}_i)\mathbf{I} + 2\boldsymbol{\beta}_i \boldsymbol{\beta}_i^\top}{(\alpha_i^2 - \boldsymbol{\beta}_i^\top \boldsymbol{\beta}_i)^2}, & \text{for } i = j, \end{cases}$$

$$\frac{\partial}{\partial \alpha_j} \nabla_{\boldsymbol{\beta}_i} f_t = \begin{cases} \mathbf{0}, & \text{for } i \neq j, \\ \frac{-4}{t} \frac{\alpha_i \boldsymbol{\beta}_i}{(\alpha_i^2 - \|\boldsymbol{\beta}_i\|_2^2)^2}, & \text{for } i = j. \end{cases}$$

The derivative of $\nabla_{\boldsymbol{\alpha}} f_t$ with respect to $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$ can be computed as

$$\nabla_{\boldsymbol{\beta}_j} \nabla_{\boldsymbol{\alpha}_i} f_t = \begin{cases} \mathbf{0}, & \text{for } i \neq j, \\ \frac{-4}{t} \frac{\alpha_i \boldsymbol{\beta}_i}{(\alpha_i^2 - \|\boldsymbol{\beta}_i\|_2^2)^2} & \text{for } i = j, \end{cases}$$

$$\frac{\partial^2 f_t}{\partial \alpha_i \partial \alpha_j} = \begin{cases} 0, & \text{for } i \neq j, \\ \frac{2}{t} \frac{\alpha_i^2 + \boldsymbol{\beta}_i^\top \boldsymbol{\beta}_i}{(\alpha_i^2 - \|\boldsymbol{\beta}_i\|_2^2)^2}, & \text{for } i = j. \end{cases}$$

Hence, the Jacobian matrix is

$$\mathbf{J} = \begin{bmatrix} \nabla_{\boldsymbol{\beta}} \nabla_{\boldsymbol{\beta}} f_t & \nabla_{\boldsymbol{\alpha}} \nabla_{\boldsymbol{\beta}} f_t \\ \nabla_{\boldsymbol{\beta}} \nabla_{\boldsymbol{\alpha}} f_t & \nabla_{\boldsymbol{\alpha}} \nabla_{\boldsymbol{\alpha}} f_t \end{bmatrix}.$$

Let $\mathbf{g} = [\nabla_{\boldsymbol{\beta}} f_t^\top, \nabla_{\boldsymbol{\alpha}} f_t^\top]^\top$. Then we have

$$\nabla_{\mathbf{y}} \mathbf{g} = [-2\mathbf{X},\, \mathbf{0}]^\top,$$

and

$$\frac{\partial g_k}{\partial X_{ij}} = \begin{cases} 2\left[\delta_{kj}(\mathbf{X}\boldsymbol{\beta} - \mathbf{y})_i + X_{ik} y_j\right], & \text{for } 1 \leq k \leq m, \\ 0, & \text{otherwise.} \end{cases}$$

As a result, the derivatives of $\boldsymbol{\beta}$ with respect to $\mathbf{y}$ and $\mathbf{X}$ is the first m rows of $-\mathbf{J}^{-1}\nabla_{\mathbf{y}}\mathbf{g}$ and $-\mathbf{J}^{-1}\nabla_{\mathbf{X}}\mathbf{g}$, respectively. With this gradient information and using the chain rule, we can obtain the gradients of our objective with respect to the response values and feature matrix. Then, we can use the projected gradient descent method described in Algorithm 4 to design our attack strategy.

3.4.2 Adversarial Attacks Against Sparse Group LASSO

Sparse group LASSO combines the ordinary and the group LASSO and exploit the sparsity and group sparsity jointly. It gives better performance when the features are formed in a group manner and only few features contribute to the response value within a group. By combining these two properties, sparse group LASSO promotes the group-wise sparsity as well as the sparsity within each group. By taking advantage of these two kinds of sparsities, sparse group LASSO helps us select more accurate features, and it has been used in climate prediction [109], heterogeneous feature representations [110], change-points estimation [48], etc. The sparse group LASSO problem tries to solve the following convex problem:

$$\min_{\boldsymbol{\beta}} \quad \|\mathbf{y} - \sum_{l=1}^{L} \mathbf{X}_l \boldsymbol{\beta}_l\|_2^2 + \lambda_1 \sum_{l=1}^{L} \sqrt{p_l} \|\boldsymbol{\beta}_l\|_2 + \lambda_2 \|\boldsymbol{\beta}\|_1. \tag{3.28}$$

Similar to problem (3.25), we assume the regression coefficients are divided into L groups and each group $\boldsymbol{\beta}_l \in \mathbb{R}^{p_l}$. In the above objective, the first term is the ordinary least square to measure the goodness of fitting, the second term promotes the group-wise sparsity, and the third term encourages the sparsity within each group.

Taking objective (3.3) into account, the design of optimal attack strategy against sparse

group LASSO can be formulated as solving a bi-level optimization problem:

$$\begin{aligned}\min_{\mathbf{y}\in\mathcal{C}_y,\mathbf{X}\in\mathcal{C}_x} \quad & \frac{1}{2}(\hat{\boldsymbol{\beta}}-\boldsymbol{\nu})^\top\mathbf{H}(\hat{\boldsymbol{\beta}}-\boldsymbol{\nu}) \\ \text{s.t.} \quad \hat{\boldsymbol{\beta}} = \operatorname*{argmin}_{\boldsymbol{\beta}} \quad & \|\mathbf{y}-\sum_{l=1}^{L}\mathbf{X}_l\boldsymbol{\beta}_l\|_2^2 \\ & + \lambda_1\sum_{l=1}^{L}\sqrt{p_l}\|\boldsymbol{\beta}_l\|_2 + \lambda_2\|\boldsymbol{\beta}\|_1.\end{aligned} \tag{3.29}$$

To solve this bi-level optimization problem, as in the previous subsection, we can transform the lower-level problem into a quadratic programming with conic and linear inequality constraints by introducing the new variables α_l for $l = 1, 2, \ldots, L$ and u_i for $i = 1, 2, \ldots, m$ as follows:

$$\operatorname*{argmin}_{\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{u}} \quad \|\mathbf{y}-\sum_{l=1}^{L}\mathbf{X}_l\boldsymbol{\beta}_l\|_2^2 + \sum_{l=1}^{L}\tilde{\lambda}_l\alpha_l + \lambda_2\sum_{i=1}^{m}u_i \tag{3.30}$$

$$\text{s.t.} \quad \|\boldsymbol{\beta}\|_2 \le \alpha_l, \quad l = 1, 2, \ldots, L, \tag{3.31}$$

$$-u_i \le \beta_i \le u_i, \quad i = 1, 2, \ldots, m, \tag{3.32}$$

where $\tilde{\lambda}_l = \lambda_1\sqrt{p}_l$. We use the similar interior-point method to solve this optimization problem. Thus, we use penalty functions for the constraints and have the new objective with a certain penalty parameter t:

$$\begin{aligned}h_t = & \|\mathbf{y}-\sum_{l=1}^{L}\mathbf{X}_l\boldsymbol{\beta}_l\|_2^2 + \sum_{l=1}^{L}\tilde{\lambda}_l\alpha_l + \lambda_2\sum_{i=1}^{m}u_i \\ & - 1/t\sum_{l=1}^{L}\log(\alpha_l^2 - \|\boldsymbol{\beta}_l\|_2^2) - 1/t\sum_{i=1}^{m}\log(u_i^2-\beta_i^2).\end{aligned}$$

The corresponding first order optimality condition is

$$\begin{cases} \nabla_{\boldsymbol{\beta}_l} h_t = & 2\mathbf{X}_l^\top(\mathbf{X}\boldsymbol{\beta}-\mathbf{y}) + 1/t \cdot \frac{2\boldsymbol{\beta_l}}{\alpha_l^2 - \|\boldsymbol{\beta}_l\|_2^2} \\ & + \frac{2\boldsymbol{\beta}_l}{t} \cdot \mathrm{diag}\Big(1/\left((u_l^1)^2 - (\beta_l^1)^2\right., \\ & \left. \ldots, 1/\left((u_l^{p_l})^2 - (\beta_l^{p_l})^2\right)\Big) = \mathbf{0}, \\ & \text{for } l = 1, 2, \ldots, L, \\ \frac{\partial h_t}{\partial \alpha_l} = & \tilde{\lambda}_l - 1/t \cdot \frac{2\alpha_l}{\alpha_l^2 - \|\boldsymbol{\beta}_l\|_2^2} = 0, \text{ for } l = 1, 2, \ldots, L, \\ \frac{\partial h_t}{\partial u_i} = & \lambda_2 - 1/t \cdot \frac{2u_i}{u_i^2 - \beta_i^2} = 0, \text{ for } i = 1, 2, \cdots, m, \end{cases}$$

where $\boldsymbol{\beta} = [\boldsymbol{\beta}_1^\top, \boldsymbol{\beta}_2^\top, \ldots, \boldsymbol{\beta}_L^\top]^\top$, $\mathbf{u} = [\mathbf{u}_1^\top, \mathbf{u}_2^\top, \ldots, \mathbf{u}_L^\top]^\top$, $\boldsymbol{\beta}_l = [\beta_l^1, \beta_l^2, \ldots, \beta_l^{p_l}]^\top$, $\mathbf{u}_l = [u_l^1, u_l^2, \ldots, u_l^{p_l}]^\top$. To use the implicit function theorem to obtain the gradient information, we need to compute the Jacobian matrix of the function on the left of the first order optimality condition. The Jacobian matrix is

$$\mathbf{J} = \begin{bmatrix} \nabla_{\boldsymbol{\beta}}\nabla_{\boldsymbol{\beta}} h_t & \nabla_{\boldsymbol{\alpha}}\nabla_{\boldsymbol{\beta}} h_t & \nabla_{\mathbf{u}}\nabla_{\boldsymbol{\beta}} h_t \\ \nabla_{\boldsymbol{\beta}}\nabla_{\boldsymbol{\alpha}} h_t & \nabla_{\boldsymbol{\alpha}}\nabla_{\boldsymbol{\alpha}} h_t & \nabla_{\mathbf{u}}\nabla_{\boldsymbol{\alpha}} h_t \\ \nabla_{\boldsymbol{\beta}}\nabla_{\mathbf{u}} h_t & \nabla_{\boldsymbol{\alpha}}\nabla_{\mathbf{u}} h_t & \nabla_{\mathbf{u}}\nabla_{\mathbf{u}} h_t \end{bmatrix},$$

where

$$\nabla_{\boldsymbol{\beta}}\nabla_{\boldsymbol{\beta}} h_t = 2\mathbf{X}^\top\mathbf{X} + \mathbf{E}_{1,1} + \mathbf{D}_{1,1},$$

in which

$$\begin{aligned} \mathbf{E}_{1,1} =& \frac{1}{t}\mathrm{diag}\left(\frac{(\alpha_1^2 - \boldsymbol{\beta}_1^\top\boldsymbol{\beta}_1)\mathbf{I} + 2\boldsymbol{\beta}_1\boldsymbol{\beta}_1^\top}{(\alpha_1^2 - \boldsymbol{\beta}_1^\top\boldsymbol{\beta}_1)^2}, \ldots, \frac{(\alpha_L^2 - \boldsymbol{\beta}_L^\top\boldsymbol{\beta}_L)\mathbf{I} + 2\boldsymbol{\beta}_L\boldsymbol{\beta}_L^\top}{(\alpha_L^2 - \boldsymbol{\beta}_L^\top\boldsymbol{\beta}_L)^2}\right), \\ \mathbf{D}_{1,1} =& 2/t \cdot \mathrm{diag}\left((u_1^2 + \beta_1^2)/(u_1^2 - \beta_1^2)^2, \ldots, (u_m^2 + \beta_m^2)/(u_m^2 - \beta_m^2)^2\right), \end{aligned}$$

$$\frac{\partial}{\partial \alpha_j}\nabla_{\boldsymbol{\beta}_i} h_t = \begin{cases} \mathbf{0}, & \text{for } i \neq j, \\ \frac{-4}{t}\frac{\alpha_i \boldsymbol{\beta_i}}{(\alpha_i^2 - \|\boldsymbol{\beta}_i\|_2^2)^2}, & \text{for } i = j, \end{cases}$$

$$\nabla_{\mathbf{u}}\nabla_{\boldsymbol{\beta}} h_t = \operatorname{diag}\Big(-4/t \cdot \frac{\beta_1 u_1}{(u_1^2 - \beta_1^2)^2}, \ldots, -4/t \cdot \frac{\beta_m u_m}{(u_m^2 - \beta_m^2)^2} \Big),$$

$$\frac{\partial^2 f_t}{\partial \alpha_i \partial \alpha_j} = \begin{cases} 0, & \text{for } i \neq j, \\ \frac{2}{t}\frac{\alpha_i^2 + \boldsymbol{\beta}_i^\top \boldsymbol{\beta}_i}{(\alpha_i^2 - \|\boldsymbol{\beta}_i\|_2^2)^2}, & \text{for } i = j, \end{cases}$$

$$\nabla_{\mathbf{u}}\nabla_{\boldsymbol{\alpha}} h_t = \mathbf{0},$$

and

$$\nabla_{\mathbf{u}}\nabla_{\mathbf{u}} h_t = \operatorname{diag}\Big(2(u_1^2 + \beta_1^2)/(u_1^2 - \beta_1^2)^2, \ldots, 2(u_m^2 + \beta_m^2)/(u_m^2 - \beta_m^2)^2 \Big).$$

Let $\mathbf{q} \triangleq [\nabla_{\boldsymbol{\beta}} h_t^\top, \nabla_{\boldsymbol{\alpha}} h_t^\top, \nabla_{\mathbf{u}} h_t^\top]^\top$, then we have

$$\nabla_{\mathbf{y}} \mathbf{q} = [-2\mathbf{X}, \mathbf{0}]^\top$$

and

$$\frac{\partial q_k}{\partial X_{ij}} = \begin{cases} 2\left[\delta_{kj}(\mathbf{X}\boldsymbol{\beta} - \mathbf{y})_i + X_{ik} y_j\right], & \text{for } 1 \leq k \leq m, \\ 0, & \text{otherwise.} \end{cases}$$

Then we have the derivative of $\boldsymbol{\beta}$ with respect to $\mathbf{y}$ being

$$\nabla_{\mathbf{y}}\boldsymbol{\beta} = -[\mathbf{J}^{-1}]_{1:m}\nabla_{\mathbf{y}}\mathbf{q}.$$

and the partial derivative of β_k with respect to $X_{i,j}$ is

$$\frac{\partial \beta_k}{\partial X_{i,j}} = \sum_{l=1}^{m} -(\mathbf{J}^{-1})_{k,l}\frac{\partial q_l}{\partial X_{i,j}}.$$

Having the gradients of $\boldsymbol{\beta}$ with respect to $\mathbf{y}$ and $\mathbf{X}$, combining the gradients of our objective with respect to $\boldsymbol{\beta}$ and using the chain rule, we can get the full gradients of our objective with respect to $\mathbf{y}$ and $\mathbf{X}$. With these gradients information, we can then employ the projected gradient descent described in Algorithm 4 to find our modification strategy.

3.5 Numerical Examples

In this section, we carry out several experiments to demonstrate the results obtained in this chapter.

3.5.1 Attack Against Ordinary LASSO

In the first numerical example, we test our algorithm on a synthetic data set. Firstly, we generate a 30×50 feature matrix $\mathbf{X}_0$. Each entry of the feature matrix is i.i.d. generated from a standard normal distribution. Then, we generate the response values, $\mathbf{y}_0$, through the model $\mathbf{y}_0 = \mathbf{X}_0\mathbf{v} + \mathbf{n}$, where $\mathbf{v}$ is the sparse vector in which only ten randomly selected positions are non-zero and each of the non-zero entry is i.i.d. drawn from the standard normal distribution; $\mathbf{n}$ is the noise vector where each entry is i.i.d. generated according to a normal distribution with zero mean and 0.1 variance. The generated dataset has Frobenius norm 38.60 of the feature matrix and ℓ_2 norm 19.26 of the response vector. Then, we set the LASSO trade-off parameter $\lambda = 2$ and use (3.7) to estimate the regression coefficients $\boldsymbol{\beta}_0$.

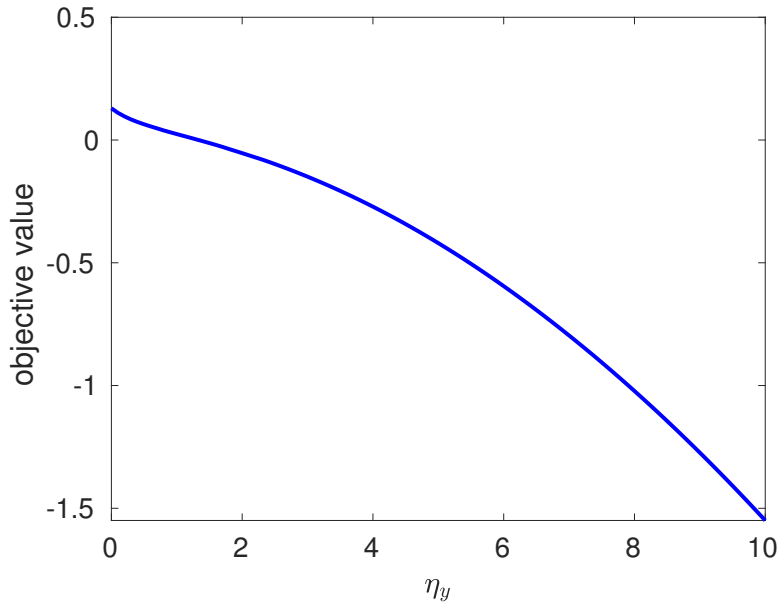


Figure 3.1: The objective value changes with the energy budget.

We randomly select one regression coefficient as the desired coefficient to be boosted and another one as the coefficient to be suppressed. In addition, we set the suppressed parameter $s_i = 1$ for $i \in S$, set boosted parameter $e_i = -1$ for $i \in E$, and set the unchanged parameter $\mu_i = 5$ for $i \in U$. We set the step-size parameter $\gamma_k = \min(\rho, \rho K_0/k)$ in Algorithm 4, where $\rho = 1$ and $K_0 = 100$.

In the first experiment, we set $\eta_x = 0$, which means that we do not modify the feature matrix and impose ℓ_2 norm constraint on the modification of the response values. Then, we vary the energy budget, η_y, to see how the energy budget influences our objective value. Fig. 3.1 illustrates that the objective value decreases as the energy budget increases, which is expected as a larger energy budget provides a larger feasible region, and thus lower objective value. Fig. 3.2 demonstrates the recovered regression coefficients when $\eta_y = 5$ along with the original regression coefficients. In the figure, 'orig' denotes the original regression coefficients, 'modi' represents the regression coefficients after our attack, 'min' is the regression coefficient we want to suppress, and 'max' denotes the regression coefficient we want to promote. As the figure demonstrates, we have successfully suppressed and promoted the corresponding coefficients while keeping other regression coefficients almost unchanged.

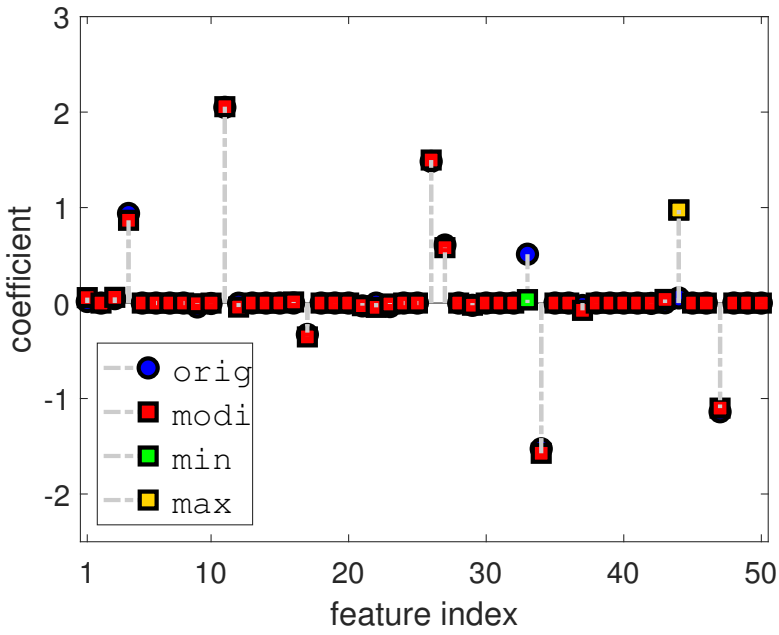


Figure 3.2: The original regression coefficients and the regression coefficients after our attacks.

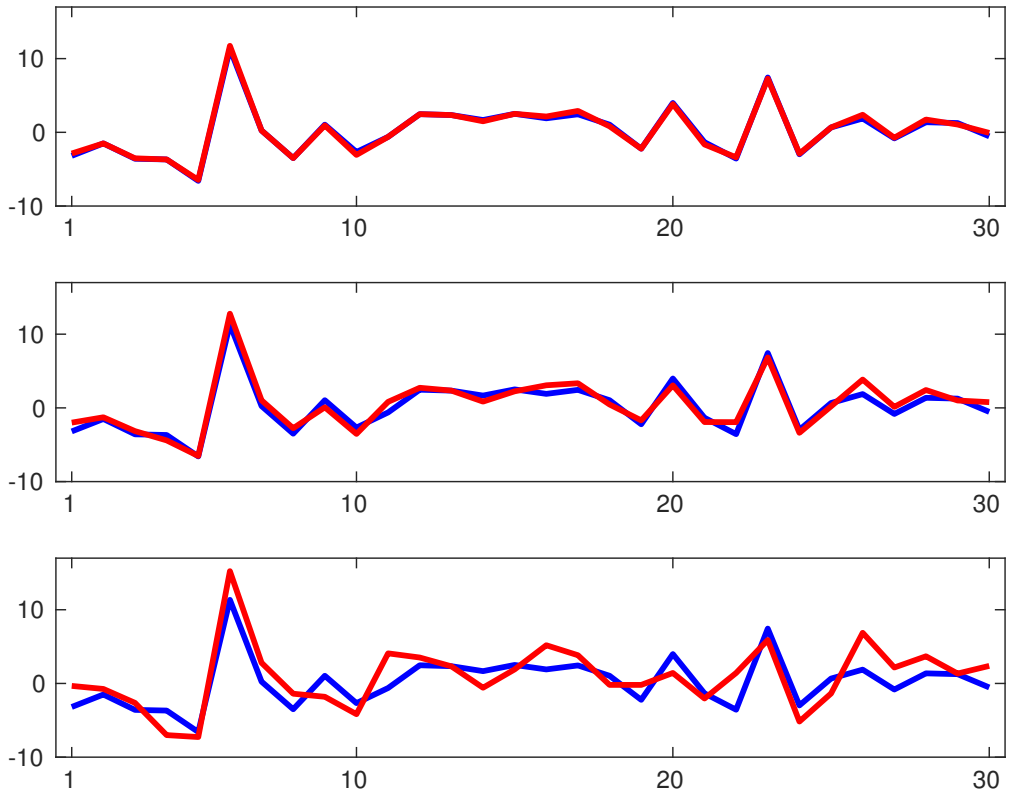


Figure 3.3: The original response values and the modified response values with different attack constraints.

In the second experiment, we also attack the response values. We fix the energy budget $\eta_y = 5$ and test different ℓ_p norm constraints on the modification of the response values as $p = 1, 2, \infty$. Fig. 3.3 shows the original and modified response values under different ℓ_p norm constraints. The x-axis denotes the index of each response value and the y-axis denotes the value of the response vector. The blue line demonstrates the original response values and the red line is the modified response values with different attack constraints. From top to bottom are the modified response values with ℓ_1, ℓ_2, and ℓ_∞ norm constraints, respectively. From the figure, we can see that the ℓ_1 norm constraint provides the smallest modification on the response values and the ℓ_∞ norm constraint provides the most significant modification, which results in objective value 0.0095 with the ℓ_1 norm constraint, objective value -0.4199 with the ℓ_2 norm constraint, and objective value -2.8813 with the ℓ_∞ norm constraint. That is because with the same radius, ℓ_1 norm ball is contained in the ℓ_2 norm ball and ℓ_2 norm ball belongs to the ℓ_∞ norm ball.

In the third experiment, we compare the modifications on the response values and on the feature matrix with the ℓ_1 constraints. First, we only attack the response values with $\eta_y = 5$, which results in objective value 0.0095. Second, we only attack the feature matrix with the same energy budget $\eta_x = 5$, which results in objective value -0.0969. Finally, we attack both the response values and the feature matrix with $\eta_y = 5$ and $\eta_x = 5$, which results in objective value -0.2291. These results indicate that both the modifications of the response values and feature matrix are effective.

In the fourth experiment, we explore the minimal energy required to suppress one regression coefficient. In this experiment, we try to make one of the non-zero coefficient to be zero while keeping other regression coefficients unchanged. Hence, we set $s_i = 1$ for $i \in S$ and $u_i = 5$ for $i \in U$. Firstly, we set $\eta_x = 0$ and only change the response values. The minimal required η_y under the ℓ_2 norm constraint to make the regression coefficient zero is recorded. Secondly, we fix $\eta_y = 0$ and only modify the feature matrix to make one regression coefficient zero. We record the minimal energy budget required for the modification of the feature

matrix in terms of the Frobenius norm. TABLE 3.1 presents the minimal energy budgets to suppress one regression coefficient. The first row is the feature index that we want to suppress. The second row denotes the coefficients before modification. The third row shows the minimal energy budget when we only modify the response values. The forth row indicates the minimal energy budget when we only modify the feature matrix. We can see from the table that the energy required to suppress the coefficient depends on the original magnitude of the coefficient. When suppressing a coefficient with a larger magnitude it requires more energy and vice versa. When we only modify the response vector, we need the energy that is about 60 ($\eta_y/\|\mathbf{y}\|_2 = 11.5/19.26 \approx 0.60$) and 9 ($\eta_y/\|\mathbf{y}\|_2 = 1.8/19.26 \approx 0.09$) percent of the ℓ_2 norm of the response vector to successfully make the largest coefficient (the 11th coefficient) and the smallest coefficient (the 17th coefficient) be zero, respectively. When only modifying the feature matrix, we need the energy that approximates to 12 ($4.5/38.6 \approx 0.12$) and 1 ($0.5/38.6 \approx 0.01$) percent of the Frobenius norm of the feature matrix to successfully make the largest and smallest coefficient be zero, respectively. This also indicates that a small perturbation of the feature matrix can suppress one regression coefficient, while a relatively larger modification of the response values is needed to suppress the same regression coefficient.

In the fifth experiment, we explore the minimal energy needed to promote one of the regression coefficients. We try different energy budgets to promote one of the regression coefficients while keeping others unchanged. So, we set $u_i = 5$ for $i \in U$ and $e_i = -1$ for i $\in E$. We record the minimal energy used to make the magnitude of one of the regression coefficients at least 0.5. The regression coefficients that we want to promote are chosen randomly among the 42 zero-valued coefficients. We randomly select 8 coefficients and TABLE 3.2 records the minimal energy. The first row indicates the feature index that we choose. The second row presents the minimal energy needed when we only modify the response vector under the ℓ_2 norm constraint. The third row shows the minimal energy needed when we only modify the feature matrix under the Frobenius norm. This table

Table 3.1: Minimal energy to suppress one regression coefficient

energy \ index	4	11	17	26	27	33	34	47
$\hat{\beta}_i$	0.9	2.1	-0.3	1.5	0.6	0.5	-1.5	-1.1
η_y	5.7	11.5	1.8	7.5	2.8	2.6	7.5	6.2
η_x	1.5	4.5	0.5	2.3	1.0	1.0	2.4	1.8

Table 3.2: Minimal energy to promote one regression coefficient

energy \ index	1	2	28	29	30	44	48	50
η_y	4.1	4.2	4.7	4.0	4.6	4.3	4.8	3.3
η_x	1.4	1.2	1.6	1.2	1.4	1.2	1.5	1.4

shows we need similar energy to promote different regression coefficients. The reason is that the original regression coefficients that we try to promote are zero-valued and we set the same magnitude, 0.5, for coefficients we try to promote. In summary, when we only modify the response values, the average minimal energy is about 22 percent of the ℓ_2 norm of the response vector. When we only modify the feature matrix, the average minimal energy is about 3 percent of the Frobenius norm of the feature matrix. This indicates that similar to the fourth experiment, we can promote one of the regression coefficients easily by modifying the feature matrix, and relatively more considerable energy is needed to modify the response values to achieve the same goal.

We now test our attack strategy using real datasets. In this task, we use the spectral intensity of the gasoline to predict its octane rating [111]. It consists of 60 samples of gasoline at 401 wavelength and their octane ratings. Fig. 3.4 provides an overview of the data samples. In this figure, the octane axis indicates the octane rating of each sample and the z-axis denotes the spectral intensities at different wavelengths. From the figure we can see that there are very high correlations among different wavelengths. When strong correlation exists among features, the learned regression coefficients are not stable and will not reveal the true important features. Thus, in the testing phase, it will result in large errors. For example, there are two perfect correlated features. Then, the two corresponding

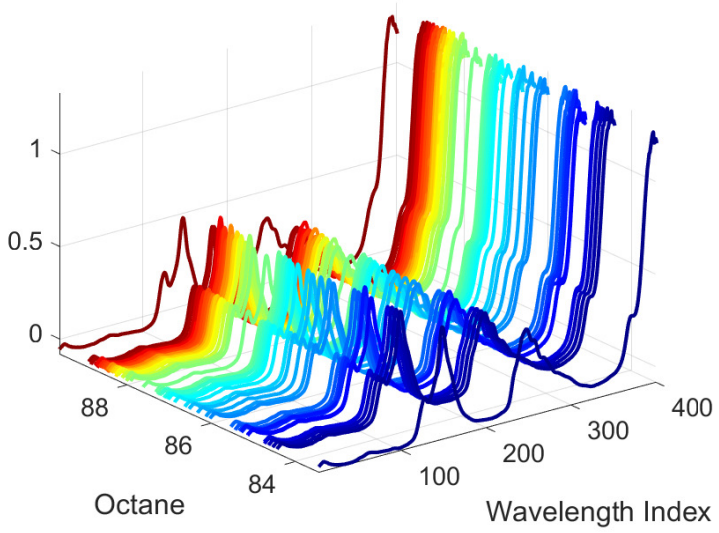


Figure 3.4: Overview of the octane data set.

regression coefficients can be any values as long as the difference of the two coefficients remains constant. Suppose the two regression coefficients are infinite large. In the testing phase, a small perturbation on one of the features will result in a huge error. Thus, we use the LASSO method to complete the regression task. We randomly choose 80% of the data samples as our training data and the rest as our test data. We do cross-validation on the training data to decide the trade-off parameter in LASSO, and it gives $\lambda = 0.5$. Using this parameter, we compute the regression coefficients. Using this regression coefficients on the test data set, we have $r^2 = 0.979$. Here, r^2 is the r-squared value and is defined as $r^2 = 1 - \|\mathbf{y} - \hat{\mathbf{y}}\|_2^2 / \|\mathbf{y} - \bar{\mathbf{y}}\|_2^2$, where $\mathbf{y}$ is the ground truth response value, $\bar{\mathbf{y}}$ is the mean value of the response value with each element being the mean of $\mathbf{y}$, and $\hat{\mathbf{y}}$ is the predicted response value. A larger r^2 value indicates better regression coefficients. The blue line in Fig. 3.5 shows the original regression coefficient. From this figure, we can see that there are several important features.

For this dataset, the Frobenius norm of the feature matrix is 20.02 and the ℓ_2 norm of the response vector is 11.75. In the next step, we modify the response values and the feature matrix with the energy budget $\eta_y = 5$ and $\eta_x = 5$ to suppress the 154th and 163th regression

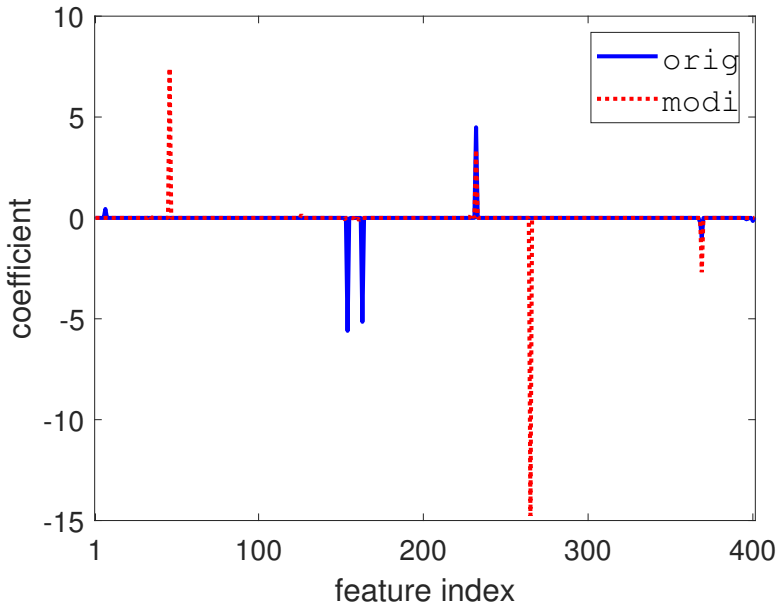


Figure 3.5: The regression coefficients before and after our attack.

coefficients, keep the 232th and 369th regression coefficients unchanged, and promote the rest of the regression coefficients. In our algorithm, we set $s_i = 1$ for $i \in S$, $e_i = -1$ for $i \in E$, $\mu_i = 50$ for $i \in U$, and step-size parameter $\gamma_k = \min(5, 5 \times 100/k)$. The red-dashed line in Fig. 3.5 shows the regression coefficients after our attacks. From the figure, we can see that we successfully promote two regression coefficients that were zero-valued before the attack. We also suppress the 154th and 163th regression coefficients and make the 232th and 369th regression coefficients change very little. Using this regression coefficients on the test data set, we got the r-squared value 0.694. Hence, by changing the response values and the feature matrix, we can easily make the system choose the wrong features.

3.5.2 Attack Against Group LASSO

In this subsection, we will employ our attack strategy on group LASSO. We will use the direction of arrival (DOA) problem as an example. In the DOA problem, we try to find the directions of the sources from the received signals of an array of sensors [112,113]. Consider a setup where the sensors are linearly located and equally spaced with half of the wavelength. Hence, the measurements of the nth sensor are $\sum_{k=1}^{K} e^{j2\pi n f_k} x_k$, where K is the number of

sources and $f_k \in (-\pi/2, \pi/2]$ is the arrival angle of the kth source. Furthermore, we assume that the number of input sources is limited. If we divide the arrival of angle equally into N grids and assume the sources are located on the grids, the DOA can be modeled as a linear signal acquisition system:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e},$$

where $\mathbf{y} \in \mathbb{C}^N$ is the measurements of the sensors, $\mathbf{A} \in \mathbb{C}^{N \times M}$, $A_{n,m} = e^{j2\pi n \frac{m-1}{M}}$, $\mathbf{x} \in \mathbb{C}^M$ is the sparse source vector where only the locations that have targets are non-zero , and $\mathbf{e} \in \mathbb{C}^N$ is the noise vector. We can first recover the sparse signal $\mathbf{x}$, and then the arrival angles can be derived from the locations of the non-zero components of $\mathbf{x}$. Further, we can solve the following LASSO problem to recover $\mathbf{x}$:

$$\underset{\mathbf{x}}{\text{argmin}} : \quad \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1, \tag{3.33}$$

where the ℓ_1 norm of $\mathbf{x}$ is defined as

$$\|\mathbf{x}\|_1 = \sum_{i=1}^{N} \sqrt{(x_i^R)^2 + (x_i^I)^2}, \tag{3.34}$$

and x_i^R and x_i^I are the real and imaginary parts of x_i, respectively. Problem (3.33) is actually a group LASSO problem if we separate its real and imaginary parts and we reformulate it as:

$$\underset{\mathbf{x}^R, \mathbf{x}^I}{\text{argmin}} \quad \|\tilde{\mathbf{y}} - \tilde{\mathbf{A}}\tilde{\mathbf{x}}\|_2^2 + \lambda \sum_{i=1}^{N} \sqrt{(x_i^R)^2 + (x_i^I)^2}, \tag{3.35}$$

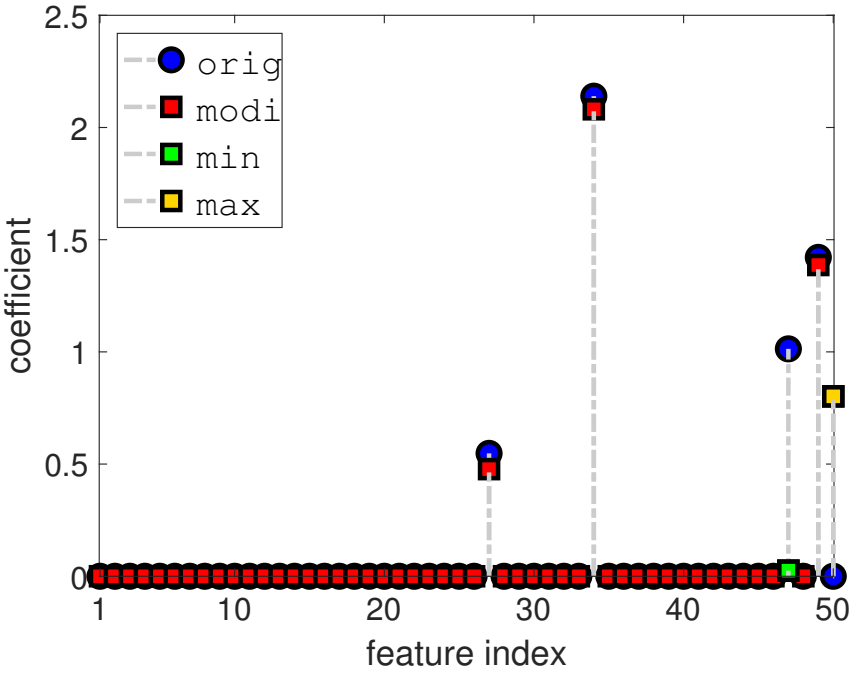


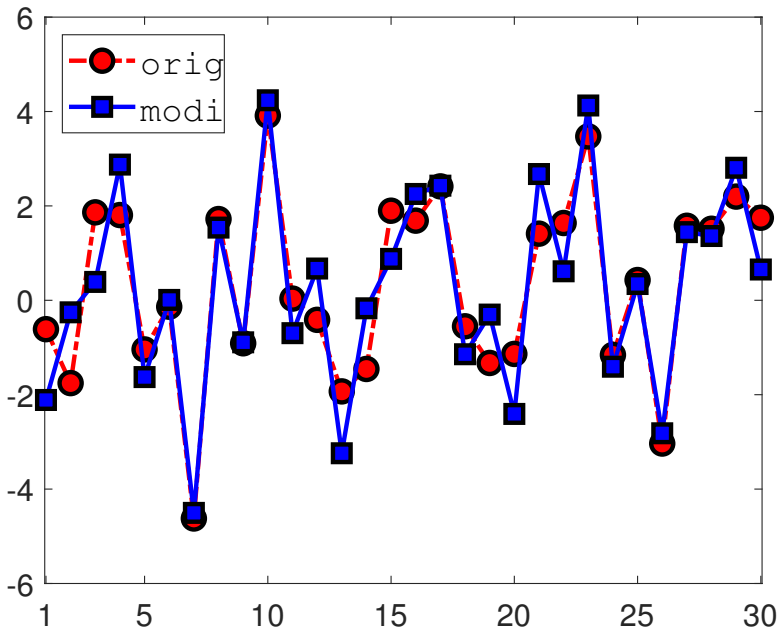
Figure 3.6: The magnitude of the coefficients before and after attacks.

where $\tilde{\mathbf{y}} = [(\mathbf{y}^R)^\top, (\mathbf{y}^I)^\top]^\top$, $\mathbf{y}^R$ and $\mathbf{y}^I$ are the real and imaginary parts of $\mathbf{y}$ respectively, $\tilde{\mathbf{x}} = [(\mathbf{x}^R)^\top, (\mathbf{x}^I)^\top]^\top$,

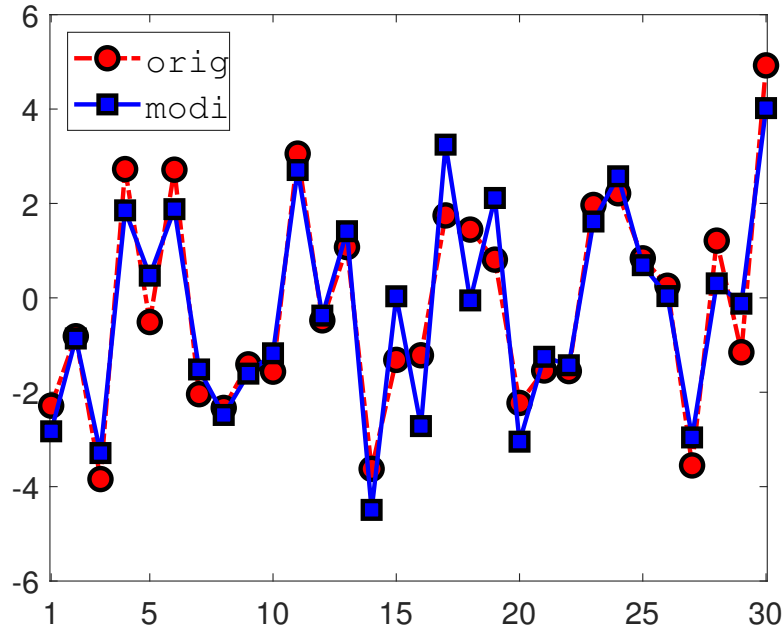
$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A}^R & \mathbf{A}^I \\ -\mathbf{A}^I & \mathbf{A}^R \end{bmatrix}, \tag{3.36}$$

and $\mathbf{A}^R$ and $\mathbf{A}^I$ are the real and imaginary parts of $\mathbf{A}$ respectively.

Since DOA is very important in military applications, in this numerical example, we demonstrate the vulnerability of DOA estimation using group LASSO. In this experiment, we assume that there are $N = 30$ sensors, $K = 4$ sources, and the sources are located in the possible $M = 50$ locations. The locations of the 4 sources are randomly chosen; for the real part and imaginary part of each signal, they are i.i.d. drawn from a standard normal distribution. The noise is i.i.d. distributed according to the standard Gaussian distribution with zero mean and 0.1 standard deviation. In our experiment, the ℓ_2 norm of $\mathbf{y}$ is 152.70, where the ℓ_2 norm of the complex vector $\mathbf{y}$ is defined as $\|\mathbf{y}\|_2 = \sqrt{\sum_{i=1}^m (y_i^I)^2 + (y_i^R)^2}$. To



(a)



(b)

Figure 3.7: The real and the imaginary part of the observed signal before and after attacks.

make our attack more practical, we only attack the measurement signal, $\mathbf{y}$. Thus, the attack process can be seen as a procedure to inject some adversarial noises into our measurements. In this attack, we set the energy of $\eta_y = 1.5$ with ℓ_∞ norm constraint and set $\lambda = 4$. We try to suppress the source on the (47)th grid with arrival of angle 306° and boost the coefficient

on the (50)th grid that originally does not have a source target. In our experiment, we set $s_i = 20$ for $i \in S$, $e_i = -1$ for $i \in E$, $\mu_i = 20$ for $i \in U$, and step-size parameter $\gamma_k = \min(1, 100/k)$.

Fig.3.6 shows the magnitude of the original regression coefficients and the regression coefficients after attack. Here, 'orig' denotes the original regression coefficients, 'modi' represents the regression coefficients after attack, 'min' and 'max' indicate the coefficients we want to supperss and boost after attack, respectively. The non-zero coefficients exactly indicate the directions of arrival of our generated target sources. The figure demonstrates that we successfully suppressed the (47)th coefficient and boost the (50)th coefficient while keeping others almost unchanged, which successfully make the receiver believe there is no target on the (47)th grid and there is a counterfeit target on the (50)th grid. Fig. 3.7 shows the real and imaginary part of the measurements before and after our attacks. Subfigure (a) represents the real part of the observed signal and subfigure (b) the imaginary part of the observed signal before and after attacks.

This figure reveals that, when we deliberately manipulate the regression coefficients in this example, the modified measurements just seem to have been perturbed by the normal noises. Hence, it is hard to detect this kind of attack.

3.5.3 Attack Against Sparse Group LASSO

In this subsection, we will use the NCEP/NCAR Reanalysis 1 dataset [114] to demonstrate our attack strategy against the sparse group LASSO based feature selection. The dataset consists of the monthly mean of temperature, sea level pressure, precipitation, relative humidity, horizontal wind speed, and vertical wind speed from 1948 to present (871 months) on the globe in a $2.5^\circ \times 2.5^\circ$ resolution. For demonstration purpose, we coarse the resolution to $10^\circ \times 10^\circ$ and we get 403 valid ocean locations. This task aims to analyze the dependencies between the records on the ocean and the records on certain land. Notably, we consider the relationship between the records on the ocean and the temperature of Brazil. Moreover,

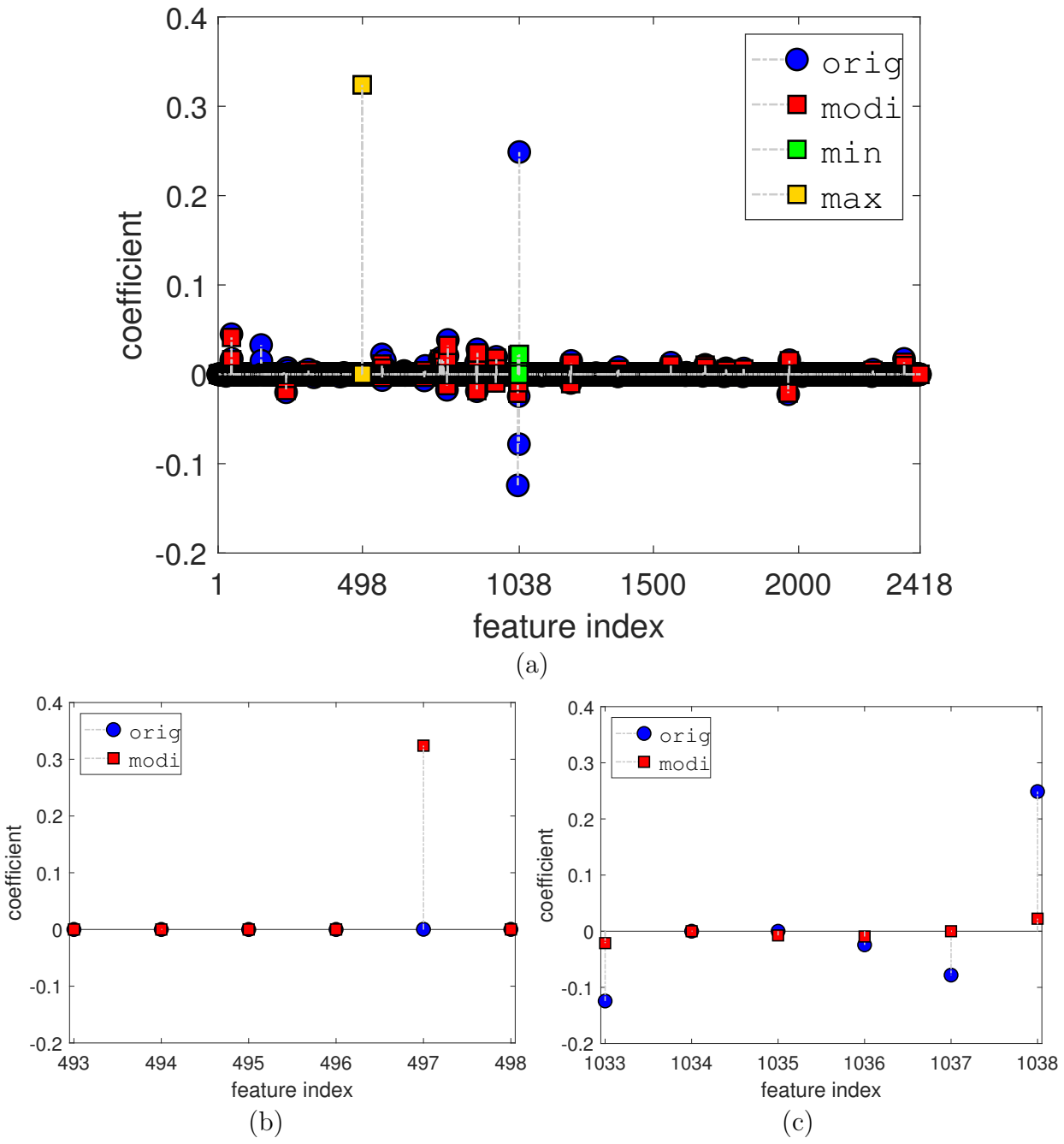


Figure 3.8: The regression coefficients before and after attacks.

we follow [109] to remove the seasonality and the trend in the data that may dominate the signal.

We use the data from Jan. 1984 to Dec. 2007 as the training data and the data from

Jan. 2008 to Dec. 2017 as test data. Hence, we have 720 training samples and 120 test samples. We use the sparse group LASSO algorithm to find the coefficients and then use these coefficients to predict the temperature of Brazil. The regression coefficients are grouped by their locations. So, each group has six coefficients. We use root mean square error (RMSE) and r-square value to measure the goodness of the regression coefficients. In this experiment, the Frobenius norm of the feature matrix is 1393.70 and the ℓ_2 norm of the response vector is 24.84. We set $\lambda_1 = \lambda_2 = N/20$, $s_i = 1$ for $i \in S$, $e_i = -1$ for $i \in E$, $\mu_i = 20$ for $i \in U$ and $\gamma_k = \min(1, 100/k)$. Our attack strategy is to use energy budgets $\eta_y = 0.2$ and $\eta_x = 0.2$ with the ℓ_∞ constraints to suppress the coefficients in group 173 and boost the coefficients in group 83 while keeping others unchanged.

Fig. 3.8 depicts the coefficients before and after our attacks. Subfigure (a) represents the regression coefficients before and after attacks. Here, 'orig' denotes the original regression coefficients, 'modi' represents the regression coefficients after Attack, 'min' and 'max' indicate the coefficients we want to suppress and boost after attack, respectively. The group coefficients that we try to maximize corresponding to the feature indices from 493 to 498 and subfigure (b) shows the coefficients in this group before and after the attack. The group coefficients we want to minimize corresponding to the feature indices from 1033 to 1038 and subfigure (c) demonstrates the coefficients in this group before and after attacks. From the figure we can see, without attack, we can find the most representative coefficients in group 173 with coordinate 40W, 20S, which is located on the ocean near the land of Brazil. After our attack, as demonstrated, we successfully suppressed the coefficients in group 173 and boosted the coefficients in group 83. By doing so, it gives us the incorrect explanation of the temperature in Brazil. Further, we get $r^2 = 0.55$ and RMSE $= 0.53$ without attack on the test data. After attack, we get $r^2 = 0.37$ and RMSE $= 0.62$ on the test data. In summary, by attacking the training data, we can manipulate the interpretation of the relationship between the features and the response value and also worsen the prediction results.

3.6 Summary

In this chapter, we have investigated the adversarial robustness of the LASSO based feature selection algorithms, including ordinary LASSO, group LASSO and sparse group LASSO. We have provided an approach to mitigate the non-differentiability of the ℓ_1 norm based feature selection methods and have designed an algorithm to obtain the optimal attack strategy. The numerical examples on synthetic data and real data have shown that feature selection based on LASSO and its variants are very vulnerable to adversarial attacks.

Chapter 4

On the Adversarial Robustness of Subspace Learning

4.1 Introduction

In this chapter, we examine the adversarial robustness of the subspace learning problem. We characterize the optimal rank-one modification strategy and the modification without any rank constraints. This chapter is organized as follows. In Chapter 4.2, we describe the precise problem formulation. In Chapter 4.3, we investigate the optimal rank-one attack strategy. We generalize our results to the case without the rank constraint in Chapter 4.4. In Chapter 4.5, we provide numerical experiments with both synthesized data and real data to illustrate results obtained in this paper. Finally, we offer concluding remarks in Chapter 4.6.

4.2 Problem Formulation

In this section, we introduce the problem formulation. Given a data matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]$ with each $\mathbf{x}_i \in \mathbb{R}^d$, our goal is to learn a low-dimension subspace via PCA. In the data matrix $\mathbf{X}$, we assume that all the preprocessing steps (such as data centering and standardization) have been done. In this chapter, we consider an adversarial setup in which an adversary will

first observe $\mathbf{X}$ and then carefully design a modification (attack) matrix $\mathbf{\Delta X}$ to change $\mathbf{X}$ to $\hat{\mathbf{X}} = \mathbf{X} + \mathbf{\Delta X}$. We denote function $g_k(\cdot)$ as the PCA operation that computes the k leading principal components. Furthermore, let $\mathbb{X} = \text{span}(g_k(\mathbf{X}))$ be a k-dimensional subspace learned from $\mathbf{X}$ and $\hat{\mathbb{X}} = \text{span}(g_k(\hat{\mathbf{X}}))$ a k-dimensional subspace learned from the modified data matrix $\hat{\mathbf{X}}$. The goal of the adversary is to design the modification matrix $\mathbf{\Delta X}$ so as to make the distance between $\mathbb{X}$ and $\hat{\mathbb{X}}$ as large as possible. To measure such a distance, we use the largest principal angle between $\mathbb{X}$ and $\hat{\mathbb{X}}$ as defined below [67].

Definition 1. *Let $\mathbb{X}$ and $\hat{\mathbb{X}}$ be two k-dimensional subspaces in $\mathbb{R}^d$. The principal angles $\{\theta_i\}_{i=1}^k$ are defined recursively:*

$$\begin{aligned} \cos(\theta_i) = \max_{\mathbf{u}_i \in \mathbb{X}, \mathbf{v}_i \in \hat{\mathbb{X}}} \quad & \mathbf{u}_i^\top \mathbf{v}_i \\ s.t. \quad & \|\mathbf{u}_i\| = \|\mathbf{v}_i\| = 1, \\ & \mathbf{u}_j^\top \mathbf{u}_i = \mathbf{v}_j^\top \mathbf{v}_i = 0, \forall\, j = 1, 2, \cdots, i-1. \end{aligned}$$

In this chapter, we will use $\|\cdot\|$ to denote the ℓ_2 norm and $\theta\big(g_k(\mathbf{X}), g_k(\hat{\mathbf{X}})\big)$ or simply θ to denote the Asimov distance between the subspace $\mathbb{X}$ estimated from $\mathbf{X}$ and the subspace $\hat{\mathbb{X}}$ estimated from $\hat{\mathbf{X}}$. Given an orthonormal basis $\mathbf{U}_{\mathbb{X}}$ of $\mathbb{X}$ and an orthonormal basis $\mathbf{U}_{\hat{\mathbb{X}}}$ of $\hat{\mathbb{X}}$, $\{\cos(\theta_1), \cdots, \cos(\theta_k)\}$ are the singular values of $\mathbf{U}_{\mathbb{X}}^\top \mathbf{U}_{\hat{\mathbb{X}}}$ [67]. Hence, the Asimov distance is determined by the smallest singular value of $\mathbf{U}_{\mathbb{X}}^\top \mathbf{U}_{\hat{\mathbb{X}}}$. It is easy to see that, if no constraint is imposed on $\mathbf{\Delta X}$, $\hat{\mathbf{X}}$ can be arbitrary and θ can be easily made to be $\pi/2$. Therefore, we impose an energy constraint on $\mathbf{\Delta X}$. In particular, we assume that the energy of $\mathbf{\Delta X}$ is less than or equal to η. In this chapter, we use the Frobenius norm $\|\mathbf{\Delta X}\|_{\text{F}}$ to measure the energy. Hence, the goal of this attacker is to solve the following optimization problem:

$$\max_{\mathbf{\Delta X}\in\mathbb{R}^{d\times n}} : \quad \theta\big(g_k(\mathbf{X}), g_k(\hat{\mathbf{X}})\big) \tag{4.1}$$
$$\text{s.t.} \quad \hat{\mathbf{X}} = \mathbf{X} + \mathbf{\Delta X},$$
$$\|\mathbf{\Delta X}\|_{\mathrm{F}} \leq \eta.$$

Even though (4.1) is a complicated non-convex optimization problem, we will fully characterize the optimal solution to (4.1) for any given η. This characterization will enable us to investigate the impact of this optimal attack with respect to the energy budget η.

Note that we consider a very powerful adversary model that has access to the whole dataset and can modify all data points. For security analysis, it is desirable to consider the worst case scenario with a powerful adversary. Furthermore, our analysis provides a universal upper bound on the maximum subspace distance incurred by any bounded energy perturbation.

4.3 Optimal Rank-one Adversarial Strategy

In this section, we will solve (4.1) for the special case where the modification matrix $\mathbf{\Delta X}$ is limited to being rank-one. The techniques and insights obtained from this special case will be useful for the general case considered in Chapter 4.4.

With this additional rank-one constraint, $\mathbf{\Delta X}$ can be written as $\mathbf{a}\mathbf{b}^\top$ for some $\mathbf{a} \in \mathbb{R}^d$ and $\mathbf{b} \in \mathbb{R}^n$, and the optimization problem (4.1) becomes

$$\max_{\mathbf{a}\in\mathbb{R}^d,\mathbf{b}\in\mathbb{R}^n} : \quad \theta\big(g_k(\mathbf{X}), g_k(\hat{\mathbf{X}})\big) \tag{4.2}$$
$$\text{s.t.} \quad \hat{\mathbf{X}} = \mathbf{X} + \mathbf{\Delta X},$$
$$\mathbf{\Delta X} = \mathbf{a}\mathbf{b}^\top,$$
$$\|\mathbf{\Delta X}\|_{\mathrm{F}} \leq \eta.$$

It is easy to see that, for any feasible solution $(\tilde{\mathbf{a}}, \tilde{\mathbf{b}})$ with $||\tilde{\mathbf{b}}|| \neq 1$, we can construct another feasible solution $(||\tilde{\mathbf{b}}||\tilde{\mathbf{a}}, \tilde{\mathbf{b}}/||\tilde{\mathbf{b}}||)$ that gives the same objective function value. Hence, without loss of optimality, we will fix the norm of $\mathbf{b}$ to be 1 throughout this section.

Based on the value of k, i.e., the dimension of the subspace we select, we will first present the solution to the case when $k = \text{rank}(\mathbf{X})$, and then generalize the result to the case when $k < \text{rank}(\mathbf{X})$.

4.3.1 Case with $k = \text{rank}(\mathbf{X})$

In this subsection, we consider the case when the dimension of the subspace selected is equal to the rank of the data matrix. In this case, the span of $\mathbf{X}$ equals the span of $g_k(\mathbf{X})$. Furthermore, we divide this case into two scenarios where the data matrix is full-rank and the data matrix is low-rank.

Full-Rank Case

In the full column rank case, $\text{rank}(\mathbf{X}) = n$, where $n \leq d$. This case arises when the number of samples is limited, for example, at the beginning of online PCA. In this case, the span of $\hat{\mathbf{X}}$ is equal to the span of $g_k(\hat{\mathbf{X}})$, and hence we can write $\theta\big(g_k(\mathbf{X}), g_k(\hat{\mathbf{X}})\big)$ as $\theta(\mathbf{X}, \hat{\mathbf{X}})$. In the following, we first find the expression of $\theta(\mathbf{X}, \hat{\mathbf{X}})$ for any given $\hat{\mathbf{X}} = \mathbf{X} + \mathbf{a}\mathbf{b}^T$. Using this expression, we then characterize the optimal attack matrix $\mathbf{\Delta X}$.

Suppose the compact SVD of $\mathbf{X}$ is $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top = \mathbf{U}\mathbf{W}$, where $\mathbf{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \cdots, \sigma_n)$. One set of orthonormal bases for the column space of $\mathbf{X}$ is $\mathbf{U}$. We can also use SVD to find a set of orthonormal bases $\tilde{\mathbf{U}}$ of $\text{span}(\hat{\mathbf{X}})$.

Since $\hat{\mathbf{X}} = \mathbf{X} + \mathbf{a}\mathbf{b}^\top$, $\tilde{\mathbf{U}}$ can be directly expressed as a function of $\mathbf{U}$ [115]:

$$\tilde{\mathbf{U}} = \mathbf{U} + (\alpha \mathbf{U}\mathbf{w} + \beta \mathbf{s})\mathbf{w}^\top,$$

where

$$\mathbf{a}_{u^\perp} = (\mathbf{I} - \mathbf{U}\mathbf{U}^\top)\mathbf{a}, \qquad \mathbf{s} = \mathbf{a}_{u^\perp}/\|\mathbf{a}_{u^\perp}\|,$$
$$\tilde{\mathbf{w}} = -\mathbf{W}^{-\top}\mathbf{b}, \qquad \mathbf{w} = \tilde{\mathbf{w}}/\|\tilde{\mathbf{w}}\|,$$
$$\omega = (1 - \mathbf{a}^\top\mathbf{U}\tilde{\mathbf{w}})/\|\mathbf{a}_{u^\perp}\|, \qquad \mathbf{g} = [\tilde{\mathbf{w}}, \omega]^\top,$$
$$\alpha = |\omega|/\|\mathbf{g}\| - 1, \qquad \beta = -\text{sign}(\omega)\|\tilde{\mathbf{w}}\|/\|\mathbf{g}\|,$$

and $\mathbf{W}^{-\top} = (\mathbf{W}^{-1})^\top$. Hence, we have $\mathbf{U}^\top\tilde{\mathbf{U}} = \mathbf{U}^\top\left(\mathbf{U} + (\alpha\mathbf{U}\mathbf{w} + \beta\mathbf{s})\mathbf{w}^\top\right) = \mathbf{I} + \alpha\mathbf{w}\mathbf{w}^\top$. The singular values of $\mathbf{I} + \alpha\mathbf{w}\mathbf{w}^\top$ are $\{1, 1, \cdots, 1 + \alpha\mathbf{w}^\top\mathbf{w}\}$. Since $\mathbf{w}^\top\mathbf{w} = 1, 1 + \alpha = |\omega|/\|\mathbf{g}\|$, the smallest singular value of $\mathbf{U}^\top\tilde{\mathbf{U}}$ is $\cos(\theta) = |\omega|/\|\mathbf{g}\|$. Our objective is to maximize θ, which is equivalent to minimizing the smallest singular value of $\mathbf{U}^\top\tilde{\mathbf{U}}$. Hence, the optimization problem (4.2) is simplified as

$$\begin{aligned} \min_{\mathbf{a},\mathbf{b}}: \quad & |\omega|/\|\mathbf{g}\| \\ \text{s.t.} \quad & \|\mathbf{a}\mathbf{b}^\top\|_{\text{F}} = \|\mathbf{a}\|\|\mathbf{b}\| \leq \eta, \end{aligned}$$

where we use the identity $\|\mathbf{a}\|\|\mathbf{b}\| = \|\mathbf{a} \cdot \mathbf{b}^\top\|_{\text{F}}$. Expanding the objective function, we have

$$\frac{|\omega|}{\|\mathbf{g}\|} = \frac{|1 + \mathbf{a}_u^\top\mathbf{W}^{-\top}\mathbf{b}|}{\|[\|\mathbf{a}_{u^\perp}\|\mathbf{W}^{-\top}\mathbf{b}, 1 + \mathbf{a}_u^\top\mathbf{W}^{-\top}\mathbf{b}]\|}, \tag{4.3}$$

where $\mathbf{a}_u = \mathbf{U}^\top\mathbf{a}$.

Since $\mathbf{W} = \mathbf{\Sigma}\mathbf{V}^\top$, we have $\mathbf{W}^{-\top}\mathbf{b} = \mathbf{\Sigma}^{-1}\mathbf{V}^\top\mathbf{b}$. As $\mathbf{V}$ is a unitary matrix, changing the coordinate $\mathbf{b} \Leftarrow \mathbf{V}^\top\mathbf{b}$ does not result in the change of the constraint. The value $\mathbf{a}_u^\top\mathbf{W}^{-\top}\mathbf{b}$ in the original coordinate is the same as $\mathbf{a}_u^\top\mathbf{\Sigma}^{-1}\mathbf{b}$ in the new coordinate. In the following, we will use this new coordinate system and the cost function in (4.3) can be written as

$$\frac{|\omega|}{\|\mathbf{g}\|} = \frac{|1 + \mathbf{a}_u^\top\mathbf{\Sigma}^{-1}\mathbf{b}|}{\|[\|\mathbf{a}_{u^\perp}\|\mathbf{\Sigma}^{-1}\mathbf{b}, 1 + \mathbf{a}_u^\top\mathbf{\Sigma}^{-1}\mathbf{b}]\|}. \tag{4.4}$$

The objective function (4.4) is zero if and only if the numerator is zero. Using the matrix norm inequality[78], we have

$$\begin{aligned}|\mathbf{a}_u^\top \boldsymbol{\Sigma}^{-1}\mathbf{b}| &\leq \|\mathbf{a}_u\|\|\mathbf{b}\|\|\boldsymbol{\Sigma}^{-1}\|_2 = \frac{1}{\sigma_n}\|\mathbf{a}_u\|\|\mathbf{b}\| \\ &\overset{(a)}{\leq} \frac{1}{\sigma_n}\|\mathbf{a}\|\|\mathbf{b}\| = \frac{1}{\sigma_n}\|\mathbf{a}\mathbf{b}^\top\|_{\mathrm{F}} \overset{(b)}{\leq} \frac{\eta}{\sigma_n},\end{aligned}$$

where $\|\boldsymbol{\Sigma}^{-1}\|_2$ is the induced 2-norm of matrix $\boldsymbol{\Sigma}^{-1}$, in (a) we use $\|\mathbf{a}_u\| \leq \|\mathbf{a}\|$, and (b) is due to the energy constraint. From the inequalities, we conclude that when $\eta < \sigma_n$, we can not make the numerator to be zero. We now consider two different cases depending on whether we can make the numerator to be zero or not.

Case 1: When $\eta > \sigma_n$, if we set

$$\mathbf{a}_u = [0, 0, \cdots, -\sigma_n]^\top, \quad \mathbf{b} = [0, 0, \cdots, 1]^\top,$$

and any $\|\mathbf{a}_{u^\perp}\|^2 = \hat{a}^2$ with $0 < \hat{a}^2 < \eta^2 - \sigma_n^2$, the numerator will be zero. Since $\mathbf{a} = \mathbf{U}\mathbf{a}_u + (\mathbf{I} - \mathbf{U}\mathbf{U}^\top)\mathbf{a}_{u^\perp}$, the attacker can make the Asimov distance to be $\pi/2$ by setting:

$$\mathbf{a} = -\sigma_n \mathbf{u}_n + \hat{a}\mathbf{u}_q, \quad \mathbf{b} = \mathbf{v}_n, \tag{4.5}$$

where $\mathbf{u}_q$ is any vector orthogonal to the column space of $\mathbf{X}$ and $0 < \hat{a}^2 < \eta^2 - \sigma_n^2$.

Case 2: When $\eta \leq \sigma_n$, the value of $1 + \mathbf{a}_u^\top \boldsymbol{\Sigma}^{-1}\mathbf{b}$ can not reach zero. In this case, it is easy to check that minimizing (4.4) is equivalent to maximizing

$$\frac{\|\mathbf{a}_{u^\perp}\|^2 \|\boldsymbol{\Sigma}^{-1}\mathbf{b}\|^2}{(1 + \mathbf{a}_u^\top \boldsymbol{\Sigma}^{-1}\mathbf{b})^2}. \tag{4.6}$$

As $\|\mathbf{b}\| = 1$, $\|\boldsymbol{\Sigma}^{-1}\mathbf{b}\|^2$ is maximized when $\mathbf{b} = [0, 0, \cdots, 1]^\top$. Furthermore, for any fixed norm of $\mathbf{a}_u$, $(1 + \mathbf{a}_u^\top \boldsymbol{\Sigma}^{-1}\mathbf{b})^2$ is minimized when $\mathbf{a}_u = [0, 0, \cdots, -\|\mathbf{a}_u\|]^\top$, $\mathbf{b} = [0, 0, \cdots, 1]^\top$.

Hence, for fixed norms of $\mathbf{a}_u$, $\mathbf{a}_{u^\perp}$, the objective function (4.6) is maximized when

$$\mathbf{a}_u = [0, 0, \cdots, -\|\mathbf{a}_u\|]^\top, \quad \mathbf{b} = [0, 0, \cdots, 1]^\top. \tag{4.7}$$

Let $c = \|\mathbf{a}_{u^\perp}\|, h = \|\mathbf{a}_u\|$. Using the optimal form of $\mathbf{a}_u$ and $\mathbf{b}$ in (4.7), the objective function (4.6) can be simplified to

$$\begin{aligned} \max_{c,h}: \quad & \frac{c^2/\sigma_n^2}{(1 - h/\sigma_n)^2} \\ \text{s.t.} \quad & (c^2 + h^2) \leq \eta^2, \end{aligned} \tag{4.8}$$

It is easy to check that the objective function is maximized when $c^2 + h^2 = \eta^2$. Hence, we have $c^2 = \eta^2 - h^2$. Inserting this value of c into the objective function and setting the derivative with respect to h to be 0, we get a unique solution $h = \eta^2/\sigma_n$. At this value of h, the second derivative is $\frac{-2\sigma_n^2}{(\sigma_n^2 - \eta^2)^3}$, which is negative. It indicates that $h = \eta^2/\sigma_n$ is indeed the maximum point. Hence, $c = \pm\eta\sqrt{1 - \eta^2/\sigma_n^2}$. This implies that the optimal solution to problem (4.2) for Case 2 is

$$\mathbf{a} = -\eta^2/\sigma_n \mathbf{u}_n \pm \eta\sqrt{1 - \eta^2/\sigma_n^2}\mathbf{u}_q, \quad \mathbf{b} = \mathbf{v}_n.$$

Summarizing the discussion above, we have the following proposition regarding the optimal value of problem (4.2) in the full-rank case.

Proposition 4.1. *In the full rank case, the optimal value of* (4.2) *is*

$$\theta^* = \begin{cases} \pi/2, & \text{if } \eta > \sigma_n \\ \arcsin(\eta/\sigma_n), & \text{if } \eta \leq \sigma_n \end{cases}.$$

Low-Rank Case

We now consider the case where $\mathbf{X}$ is not full rank. Let $k < \min(d, n)$ be the rank of $\mathbf{X}$. In this subsection, with a slight abuse of notation, we write the full SVD of $\mathbf{X}$ as $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$. The optimal attack matrix could be found by solving

$$\begin{aligned} \max_{\mathbf{a}\in\mathbb{R}^d, \mathbf{b}\in\mathbb{R}^n} : \quad & \theta\big(\mathbf{X}, g_k(\hat{\mathbf{X}})\big) \\ \text{s.t.} \quad & \hat{\mathbf{X}} = \mathbf{X} + \mathbf{a}\mathbf{b}^\top, \\ & \|\mathbf{a}\|\|\mathbf{b}\| \le \eta. \end{aligned} \tag{4.9}$$

We can further simplify this optimization problem as

$$\begin{aligned} \max_{\mathbf{a}\in\mathbb{R}^{k+1}, \mathbf{b}\in\mathbb{R}^{k+1}} : \quad & \theta\big(\tilde{\mathbf{\Sigma}}, g_k(\mathbf{Y})\big) \\ \text{s.t.} \quad & \mathbf{Y} = \tilde{\mathbf{\Sigma}} + \mathbf{a}\mathbf{b}^\top, \\ & \|\mathbf{a}\|\|\mathbf{b}\| \le \eta, \end{aligned} \tag{4.10}$$

where $\tilde{\mathbf{\Sigma}} = \mathrm{diag}(\sigma_1, \sigma_2, \cdots, \sigma_k, 0)$ and $\{\sigma_1, \sigma_2, \cdots, \sigma_k\}$ are singular values of $\mathbf{X}$. Detailed proof of the equivalence between (4.9) and (4.10) can be found in Appendix B. Here, we describe the main idea of the proof. The primary step of the simplification is to left multiply the unitary matrix $\mathbf{U}^\top$ and right multiply the unitary matrix $\mathbf{V}$ on both $\mathbf{X}$ and $\hat{\mathbf{X}}$. Note that multiplying a unitary matrix does not change the column space and its singular values. In addition, a rank-one modification can only add at most one principal component orthogonal to its original column subspace. Hence, by changing the coordinates, $\mathbf{a}$ and $\mathbf{b}$ are $k+1$ dimensional vectors.

To solve problem (4.10), we divide it into two cases based on the value of the energy budget.

Case 1: When $\eta > \sigma_k$, it is simple to verify that the solution
$\mathbf{a} = [0, 0, \cdots, \eta]^\top$, $\mathbf{b} = [0, 0, \cdots, 1]^\top$ leads to the maximal Asimov distance, which is $\pi/2$.

Case 2: When $\eta \leq \sigma_k$, the following theorem characterizes the form of optimal $\mathbf{a}$ and $\mathbf{b}$.

Theorem 4.1. *There exists an optimal solution to problem* (4.10) *in the following form*

$$\mathbf{a} = [0, \cdots, 0, a_k, a_{k+1}]^\top, \mathbf{b} = [0, 0, \cdots, 0, 1, 0]^\top, \tag{4.11}$$

with $a_k^2 + a_{k+1}^2 = \eta^2$.

Proof. Please see Appendix C. □

In the following, we will find the optimal values of a_k and a_{k+1}. Since $\|\mathbf{a}\|^2 = \eta^2$ and $\mathbf{a}$ is in the form of (4.11), we can write $\mathbf{a} = \eta[0, 0, \cdots, \cos(\alpha), \sin(\alpha)]^\top$, where $\alpha \in [0, 2\pi)$. To compute the k leading principal components of $\mathbf{Y}$, we can perform the eigenvalue decomposition of $\mathbf{Y}\mathbf{Y}^\top$,

$$\mathbf{Y}\mathbf{Y}^\top = \begin{bmatrix} \mathbf{\Lambda}_{k-1}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{c}\mathbf{c}^\top \end{bmatrix},$$

where $\mathbf{c} = [\sigma_k + \eta\cos\alpha, \eta\sin(\alpha)]^\top$, $\mathbf{\Lambda}_{k-1} = \mathrm{diag}(\sigma_1, \sigma_2, \cdots, \sigma_{k-1})$. Suppose the compact SVD of $\mathbf{Y}\mathbf{Y}^\top$ is $\mathbf{Y}\mathbf{Y}^\top = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}^\top$, where

$$\hat{\mathbf{U}} = \begin{bmatrix} \mathbf{I}_{k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{z} \end{bmatrix},$$

and $\mathbf{z} \in \mathbb{R}^2$ is the eigenvector of $\mathbf{c}\mathbf{c}^\top$ corresponding to its nonzero eigenvalue. Since one orthonormal basis of $\mathrm{span}(\tilde{\mathbf{\Sigma}})$ is $[\mathbf{I}_k, \mathbf{0}]^\top$, the Asimov distance is determined by the singular values of

$$\begin{bmatrix} \mathbf{I}_k \\ \mathbf{0} \end{bmatrix}^\top \cdot \begin{bmatrix} \mathbf{I}_{k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{z} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{k-1} & \mathbf{0} \\ \mathbf{0} & z_1 \end{bmatrix}.$$

Hence, the Asimov distance is $\arccos(|z_1|)$. Since $\mathbf{c}$ is the eigenvector of $\mathbf{c}\mathbf{c}^\top$ corresponding

to its nonzero eigenvalue, we have $|z_1| = \frac{|c_1|}{\|\mathbf{c}\|}$. Our objective function is reduced to

$$\min_{\alpha\in[0,2\pi)} : \quad \frac{|\sigma_k + \eta\cos(\alpha)|}{\|[\sigma_k + \eta\cos(\alpha), \eta\sin(\alpha)]\|}. \tag{4.12}$$

It is simple to show that the optimal solution to (4.12) is

$$\alpha^* = \arccos(-\eta/\sigma_k) \tag{4.13}$$

or

$$\alpha^* = 2\pi - \arccos(-\eta/\sigma_k). \tag{4.14}$$

Substitute the optimal solution of α^* in (4.13) or (4.14) into the objective of problem (4.12), we have $\sin(\theta^*) = \eta/\sigma_k$. Hence, the optimal solution to problem (4.10) is

$$\mathbf{a} = \left[0, 0, \cdots, -\eta^2/\sigma_k, \pm\eta\sqrt{1-\eta^2/\sigma_k^2}\right]^\top,$$
$$\mathbf{b} = [0, 0, \cdots, 0, 1, 0]^\top,$$

which indicates that the optimal solution to problem (4.9) is

$$\mathbf{a} = -\eta^2/\sigma_k\mathbf{u}_k \pm \eta\sqrt{1-\eta^2/\sigma_k^2}\mathbf{u}_q, \quad \mathbf{b} = \mathbf{v}_k,$$

where $\mathbf{u}_q$ is any vector orthogonal to the column space of $\mathbf{X}$. The corresponding optimal subspace distance is $\theta^* = \arcsin(\eta/\sigma_k)$. In summary, we have

Proposition 4.2. *The optimal Asimov distance in the low-rank case is*

$$\theta^* = \begin{cases} \pi/2, & \text{if } \eta > \sigma_k \\ \arcsin(\eta/\sigma_k), & \text{if } \eta \le \sigma_k \end{cases}. \tag{4.15}$$

The result is similar to the full column rank case characterized in Proposition 4.1.

4.3.2 Case with $k < \mathbf{rank}(\mathbf{X})$

In this section, we consider the more practical but much more challenging case with $k < \text{rank}(\mathbf{X})$.

Given the data matrix $\mathbf{X} \in \mathbb{R}^{d\times n}$, without loss of generality, we assume $d \leq n$ and $\text{rank}(\mathbf{X}) = d$. Assume the full SVD of $\mathbf{X}$ is $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, where $\mathbf{U} \in \mathbb{R}^{d\times d}$, $\mathbf{\Sigma} \in \mathbb{R}^{d\times n}$, $\mathbf{V} \in \mathbb{R}^{n\times n}$, and the singular values of $\mathbf{X}$ are $\{\sigma_1, \sigma_2, \cdots, \sigma_k, \cdots, \sigma_d\}$. Recall that we denote $g_k(\cdot)$ as the PCA operation that computes the k leading principal components. In this scenario, as the original data matrix is not low-rank, we will perform PCA both on the original data matrix and on the modified data matrix. Hence, the optimal rank-one modification matrix can be found by solving the following optimization problem

$$\begin{aligned} \max_{\mathbf{a}\in\mathbb{R}^d, \mathbf{b}\in\mathbb{R}^n} \quad & \theta\big(g_k(\mathbf{X}), g_k(\hat{\mathbf{X}})\big) \\ \text{s.t.} \quad & \hat{\mathbf{X}} = \mathbf{X} + \mathbf{a}\mathbf{b}^\top, \\ & \|\mathbf{a}\mathbf{b}^\top\|_{\text{F}} \leq \eta. \end{aligned} \tag{4.16}$$

By diagonalizing the data matrix and using similar arguments in Appendix B, (4.16) can be further simplified as

$$\begin{aligned} \max_{\mathbf{a}\in\mathbb{R}^d, \mathbf{b}\in\mathbb{R}^n} \quad & \theta\big(g_k(\mathbf{\Sigma}), g_k(\mathbf{Y})\big) \\ \text{s.t.} \quad & \mathbf{Y} = \mathbf{\Sigma} + \mathbf{a}\mathbf{b}^\top, \\ & \|\mathbf{a}\mathbf{b}^\top\|_{\text{F}} \leq \eta, \end{aligned} \tag{4.17}$$

where $g_k(\mathbf{\Sigma}) = [\mathbf{I}_k, \mathbf{0}]^\top \in \mathbb{R}^{d\times k}$. Here we also perform variable change $\mathbf{a} \Leftarrow \mathbf{U}^\top\mathbf{a}$ and $\mathbf{b} \Leftarrow \mathbf{V}^\top\mathbf{b}$. To solve this optimization problem, we divide it into two cases depending on the energy budget and the difference between σ_k and σ_{k+1}.

Case 1: When $\eta \geq \sigma_k - \sigma_{k+1}$, we have one simple solution $\mathbf{a} = [0, 0, \cdots, 0, \eta, 0, \cdots, 0]^\top$, where η is in the $(k+1)$th coordinate, and $\mathbf{b} = [0, 0, \cdots, 0, 1, 0, \cdots, 0]^\top$, where element 1 is

in the $(k+1)$th coordinate. Clearly, this setting of $\mathbf{a}$ and $\mathbf{b}$ leads to the maximal subspace distance, which is $\pi/2$.

Case 2: When $\eta < \sigma_k - \sigma_{k+1}$, the following theorem gives the form of the optimal solution.

Theorem 4.2. *The optimal solution to problem* (4.17) *should be in the form of*

$$\mathbf{a} = [0, 0, \cdots, a_k, a_{k+1}, 0, \cdots, 0]^\top, \tag{4.18}$$

$$\mathbf{b} = [0, 0, \cdots, b_k, b_{k+1}, 0, \cdots, 0]^\top, \tag{4.19}$$

where $a_k^2 + a_{k+1}^2 = \eta^2$ *and* $b_k^2 + b_{k+1}^2 = 1$.

Proof. Please see Appendix D for details. □

As the optimal solution of $\mathbf{a}$ and $\mathbf{b}$ is in the form of (4.18) and (4.19), we can parametrize $\mathbf{a}$ and $\mathbf{b}$ with parameters α and β using $\mathbf{a} = \eta[0, 0, \cdots, \cos(\alpha), \sin(\alpha), 0, \cdots, 0]^\top$ and $\mathbf{b} = [0, 0, \cdots, \cos(\beta), \sin(\beta), 0, \cdots, 0]^\top$ respectively.

As a result, the modified data matrix $\mathbf{Y}$ can be written as

$$\mathbf{Y} = \begin{bmatrix} \mathbf{\Sigma}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{\Sigma}_3 & \mathbf{0} \end{bmatrix},$$

where $\mathbf{\Sigma}_1 = \mathrm{diag}(\sigma_1, \sigma_2, \cdots, \sigma_{k-1})$, $\mathbf{\Sigma}_3 = \mathrm{diag}(\sigma_{k+2}, \cdots, \sigma_d)$, and

$$\mathbf{\Sigma}_2 = \begin{bmatrix} \sigma_k + \eta\cos(\alpha)\cos(\beta) & \eta\cos(\alpha)\sin(\beta) \\ \eta\sin(\alpha)\cos(\beta) & \sigma_{k+1} + \eta\sin(\alpha)\sin(\beta) \end{bmatrix}. \tag{4.20}$$

Since $\mathbf{Y}$ has the pseudo block diagonal form, the singular values and principal components of $\mathbf{Y}$ are determined by the SVD of $\mathbf{\Sigma}_1$, $\mathbf{\Sigma_2}$, and $\mathbf{\Sigma}_3$. For notation convenience, we denote $\mathbf{\Sigma_2} = \mathbf{D} + \eta\bar{\mathbf{a}}\bar{\mathbf{b}}^\top$, where $\mathbf{D} = \mathrm{diag}(\sigma_k, \sigma_{k+1})$, $\bar{\mathbf{a}} = [\cos\alpha, \sin\alpha]^\top$, and $\bar{\mathbf{b}} = [\cos\beta, \sin\beta]^\top$. Let ξ_1 and ξ_2 be the two singular values of $\mathbf{\Sigma}_2$ and denote their corresponding left singular

vectors as

$$\mathbf{W} = [\mathbf{w}_1\, \mathbf{w}_2] = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix}. \tag{4.21}$$

The following lemma characterizes the form of the k-dimensional subspace learned by PCA from $\mathbf{Y}$.

Lemma 4.1.

$$g_k(\mathbf{Y}) = \begin{bmatrix} \mathbf{I}_{k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{w}_1 \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

Proof. According to the perturbation theory [116], the singular values of $\mathbf{\Sigma}_2$ must satisfy

$$\xi_2 < \sigma_k, \quad \xi_1 > \sigma_{k+1}.$$

It indicates that $\xi_1 > \sigma_k$, $\xi_2 > \sigma_k$ and $\xi_1 < \sigma_{k+1}$, $\xi_2 < \sigma_{k+1}$ will not happen. Hence, we will select the eigenvector corresponding to singular value ξ_1 as one of the leading k principal components, which completes the proof. □

Since one set of orthonormal bases for $g_k(\mathbf{\Sigma})$ is $[\mathbf{I}_k, \mathbf{0}]^\top$, the subspace distance $\theta\big(g_k(\mathbf{\Sigma}), g_k(\mathbf{Y})\big)$ is determined by the singular values of

$$\begin{bmatrix} \mathbf{I}_k \\ \\ \mathbf{0} \end{bmatrix}^\top \cdot \begin{bmatrix} \mathbf{I}_{k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{w}_1 \\ \mathbf{0} & \mathbf{0} \end{bmatrix} = \operatorname{diag}(1, 1, \cdots, \cos\varphi).$$

Hence, the subspace distance is $\arccos(|\cos\varphi|)$ and our optimization problem can be equiv-

alently formulated as

$$\min_{\alpha\in[0,2\pi),\beta\in[0,2\pi)} \quad |\cos\varphi|. \tag{4.22}$$

Let $\mathbf{Z} = \mathbf{\Sigma}_2\mathbf{\Sigma}_2^\top$, we can compute $\mathbf{W}$ through eigenvalue decomposition of $\mathbf{Z}$. According to the equality $\mathbf{\Sigma}_2\mathbf{\Sigma}_2^\top = \mathbf{W}\cdot\mathrm{diag}(\xi_1^2,\xi_2^2)\cdot\mathbf{W}^\top$, we have

$$\begin{aligned}\mathbf{Z} &= \begin{bmatrix} Z_{1,1} & Z_{1,2} \\ Z_{2,1} & Z_{2,2} \end{bmatrix} \\ &= \begin{bmatrix} \xi_1^2\cos^2\varphi + \xi_2^2\sin^2\varphi & (\xi_1^2-\xi_2^2)\cos\varphi\sin\varphi \\ (\xi_1^2-\xi_2^2)\cos\varphi\sin\varphi & \xi_1^2\sin^2\varphi + \xi_2^2\cos^2\varphi \end{bmatrix}.\end{aligned}$$

From this equation, we obtain

$$\begin{cases} \cos(2\varphi)(\xi_1^2-\xi_2^2) = Z_{1,1} - Z_{2,2} \\ \sin(2\varphi)(\xi_1^2-\xi_2^2) = Z_{1,2} + Z_{2,1} \end{cases}.$$

Then we can compute φ through

$$\varphi = 0.5\mathrm{atan2}(a_y, a_x), \tag{4.23}$$

where $\mathrm{atan2}(\cdot,\cdot)$ is the four-quadrant inverse tangent function, $a_x = Z_{1,1} - Z_{2,2}$, and $a_y = Z_{1,2} + Z_{2,1}$. In our case, the specific expressions of a_x and a_y are

$$\begin{cases} a_x &= \sigma_k^2 - \sigma_{k+1}^2 + 2\sigma_k\eta\cos(\alpha)\cos(\beta) - 2\sigma_{k+1}\eta\sin(\alpha)\sin(\beta) + \eta^2\cos(2\alpha), \\ a_y &= 2\eta\Big(\sigma_k\sin(\alpha)\cos(\beta) + \sigma_{k+1}\cos(\alpha)\sin(\beta) + \eta\cos(\alpha)\sin(\alpha)\Big). \end{cases} \tag{4.24}$$

Let us write a_x and a_y as a function of α and β: $a_x = a_x(\alpha,\beta)$ and $a_y = a_y(\alpha,\beta)$. To

further restrict the domains of α and β, we analyze the properties of the angle φ in (4.23) as a function of α and β. First, we have $a_x(\alpha,\beta) = a_x(\pi+\alpha,\pi+\beta)$ and $a_y(\alpha,\beta) = a_y(\pi+\alpha,\pi+\beta)$. So $\varphi(\alpha,\beta) = \varphi(\pi+\alpha,\pi+\beta)$. This property indicates that we only need to consider the function value in the domain $\alpha \in [0,\pi], \beta \in [-\pi,\pi]$. Second, $a_x(\alpha,\beta) = a_x(\pi-\alpha,\pi-\beta)$ and $a_y(\alpha,\beta) = -a_y(\pi-\alpha,\pi-\beta)$, and then we have $\varphi(\alpha,\beta) = -\varphi(\pi-\alpha,\pi-\beta)$. Since $\cos(\varphi)$ is an even function, we only need to consider the function with domain $\alpha \in [0,\pi/2], \beta \in [-\pi,\pi]$. Note that $\mathbf{\Sigma}_2$ is in the form of (4.20), the variance in the direction of $\mathbf{e}_k$ is $v_k = \cos(\alpha)^2 + \sigma_k^2 + 2\cos(\alpha)\cos(\beta)$, and the variance in the direction of $\mathbf{e}_{k+1}$ is $v_{k+1} = \sin(\alpha)^2 + \sigma_{k+1}^2 + 2\sin(\alpha)\sin(\beta)$. To maximize the subspace distance, we should make v_k small and make v_{k+1} large. Apparently, the sign of $\cos(\alpha)\cos(\beta)$ should be negative and the sign of $\sin(\alpha)\sin(\beta)$ should be positive. Hence, the optimal α and β should satisfy $\alpha \in [0,\pi/2]$ and $\beta \in [\pi/2,\pi]$. As a result, the optimization problem (4.22) can be written as

$$\min_{\alpha\in[0,\pi/2],\beta\in[\pi/2,\pi]} : \ |\cos\left(\varphi(\alpha,\beta)\right)|. \tag{4.25}$$

The following theorem characterizes the optimal solution to problem (4.25).

Theorem 4.3. *The optimal solution to problem* (4.25) *is*

$$\begin{cases} \alpha^* &= \arccos\left(\sqrt{\frac{\sigma_k^2-\sigma_{k+1}^2+\eta^2-\sqrt{H}}{2(\sigma_k^2-\sigma_{k+1}^2)}}\right), \\ \beta^* &= \arccos\left(-\sqrt{\frac{\sigma_k^2-\sigma_{k+1}^2+\eta^2+\sqrt{H}}{2(\sigma_k^2-\sigma_{k+1}^2)}}\right), \end{cases} \tag{4.26}$$

where $H = \sigma_k^4 + \sigma_{k+1}^4 + \eta^4 - 2\sigma_k^2\sigma_{k+1}^2 - 2\sigma_k^2\eta^2 - 2\sigma_{k+1}^2\eta^2$.

Proof. Please see Appendix E. □

Accordingly, the optimal solution to problem (4.16) is

$$\mathbf{a}^* = \eta\cos(\alpha^*)\mathbf{u}_k + \eta\sin(\alpha^*)\mathbf{u}_{k+1}, \tag{4.27}$$

$$\mathbf{b}^* = \cos(\beta^*)\mathbf{v}_k + \sin(\beta^*)\mathbf{v}_{k+1}. \tag{4.28}$$

Furthermore, the optimal subspace distance θ^* can be computed according to (4.24) and (4.23). Moreover, according to the properties of the function $\varphi(\alpha, \beta)$ we have discussed before, there are other three optimal solutions

$$(-\alpha^*, -\beta^*), \quad (\pi - \alpha^*, \pi - \beta^*), \quad (\alpha^* - \pi, \beta^* - \pi),$$

which lead to the same optimal objective value.

4.4 Optimal Adversarial Strategy without the Rank Constraint

Using the insights gained from Chapter 4.3, we now characterize the optimal attack strategy in the general case without the rank-one constraint by solving (4.1). We will directly consider the general case with $k \leq \text{rank}(\mathbf{X})$.

Following the similar transformation from (4.9) to (4.10), we can simplify the optimization problem (4.1) as

$$\begin{aligned} \max_{\mathbf{B} \in \mathbb{R}^{d \times n}} : \quad & \theta\big(g_k(\mathbf{\Sigma}), g_k(\mathbf{Y})\big) \\ \text{s.t.} \quad & \mathbf{Y} = \mathbf{\Sigma} + \mathbf{B}, \\ & \|\mathbf{B}\|_{\text{F}} \leq \eta, \end{aligned} \tag{4.29}$$

where without loss of generality we assume $d \leq n$, the full SVD of the data matrix is $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, the singular values of the data matrix are $\{\sigma_1, \sigma_2, \cdots, \sigma_d\}$, and $\mathbf{B} = \mathbf{U}^\top \mathbf{\Delta} \mathbf{X} \mathbf{V}$. To identify the optimal modification matrix $\mathbf{B}$ in problem (4.29), we divide it into two cases. **Case 1:** When $\eta \geq \frac{\sigma_k - \sigma_{k+1}}{\sqrt{2}}$, by setting $b_{k,k} = -\eta/\sqrt{2}$, $b_{k+1,k+1} = \eta/\sqrt{2}$, and all other entries of $\mathbf{B}$ to zero, where $b_{i,j}$ is the element in the ith row and jth column of $\mathbf{B}$, it will lead to the maximal subspace distance, $\pi/2$.

Case 2: When $\eta < \frac{\sigma_k - \sigma_{k+1}}{\sqrt{2}}$, the following theorem states the form of the optimal $\mathbf{B}$.

Theorem 4.4. *The optimal* $\mathbf{B}$ *to problem* (4.29) *has only four possible non-zero entries:* $b_{k,k}, b_{k,k+1}, b_{k+1,k}$ *and* $b_{k+1,k+1}$.

Proof. Please see Appendix F. □

This characterization reduces the complexity of problem (4.29). Using this optimal form of $\mathbf{B}$ and following similar steps leading to (4.23), we can write the subspace distance as

$$\theta = 0.5 \left| \text{atan2}(b_y, b_x) \right|, \tag{4.30}$$

where

$$\begin{aligned} b_y &= 2\big((b_{k,k} + \sigma_k) b_{k+1,k} + (b_{k+1,k+1} + \sigma_{k+1}) b_{k,k+1}\big), \\ b_x &= (b_{k,k} + \sigma_k)^2 + b_{k,k+1}^2 - (b_{k+1,k+1} + \sigma_{k+1})^2 - b_{k+1,k}^2. \end{aligned}$$

It is easy to see that we can change the sign of b_y by changing the signs of $b_{k,k+1}$ and $b_{k+1,k}$. We also have $b_x > 0$, as

$$\begin{aligned} &\frac{b_x}{\|[b_{k,k} + \sigma_k,\, b_{k,k+1}]\| + \|[b_{k+1,k+1} + \sigma_{k+1},\, b_{k+1,k}]\|} \\ &= \|[b_{k,k} + \sigma_k,\, b_{k,k+1}]\| - \|[b_{k+1,k+1} + \sigma_{k+1},\, b_{k+1,k}]\| \\ &\geq \sigma_k - \sigma_{k+1} - \|[b_{k,k},\, b_{k,k+1}]\| - \|[b_{k+1,k},\, b_{k+1,k+1}]\| \\ &\geq \sigma_k - \sigma_{k+1} - \sqrt{2}\eta > 0. \end{aligned}$$

Using these two facts and the fact that $\text{atan2}(b_y, b_x)$ is an odd function of b_y when $b_x > 0$, we know that maximizing θ in (4.30) is equivalent to maximizing b_y/b_x. Hence, our optimization

problem can be written as

$$\begin{aligned} \max_{\mathbf{u}}: \quad & \frac{\mathbf{u}^\top \mathbf{A}_1 \mathbf{u}}{\mathbf{u}^\top \mathbf{A}_2 \mathbf{u}} \\ \text{s.t.} \quad & \|\mathbf{u} - \boldsymbol{\sigma}\|^2 \leq \eta^2, \end{aligned} \tag{4.31}$$

where $\mathbf{u} \triangleq \bar{\mathbf{b}} + \boldsymbol{\sigma}$ with $\bar{\mathbf{b}} = [b_{k,k}, b_{k+1,k}, b_{k,k+1}, b_{k+1,k+1}]^\top$ and $\boldsymbol{\sigma} = [\sigma_k, 0, 0, \sigma_{k+1}]^\top$,

$$\mathbf{A}_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \text{and} \quad \mathbf{A}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}.$$

The objective function is the ratio of two quadratic functions. It is a non-convex problem in general. In the following, we transform this problem into a feasibility problem and obtain the closed-form solution analytically.

Let λ denote the value of the objective function in (4.31). We can rewrite the optimization problem (4.31) as

$$\begin{aligned} \max_{\lambda, \mathbf{u}}: \quad & \lambda \\ \text{s.t.} \quad & \frac{\mathbf{u}^\top \mathbf{A}_1 \mathbf{u}}{\mathbf{u}^\top \mathbf{A}_2 \mathbf{u}} = \lambda, \\ & \|\mathbf{u} - \boldsymbol{\sigma}\|^2 \leq \eta^2. \end{aligned} \tag{4.32}$$

The first constraint can be written as $\mathbf{u}^\top (\mathbf{A}_1 - \lambda \mathbf{A}_2)\mathbf{u} = 0$, where

$$\begin{bmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} \end{bmatrix} \triangleq \mathbf{A}_1 - \lambda \mathbf{A}_2 = \begin{bmatrix} -\lambda & 1 & 0 & 0 \\ 1 & \lambda & 0 & 0 \\ 0 & 0 & -\lambda & 1 \\ 0 & 0 & 1 & \lambda \end{bmatrix}.$$

To further simplify the constraint, we perform eigenvalue decomposition on $\mathbf{Q} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^\top$, where $\mathbf{\Lambda} = \text{diag}(\sqrt{\lambda^2+1}, -\sqrt{\lambda^2+1})$ and

$$\mathbf{P} = t \begin{bmatrix} 1 & -(\sqrt{\lambda^2+1}+\lambda) \\ \sqrt{\lambda^2+1}+\lambda & 1 \end{bmatrix}, \tag{4.33}$$

with $t = 1/\sqrt{(\sqrt{\lambda^2+1}+\lambda)^2+1}$.

We further perform variable change $\mathbf{v} \triangleq \text{diag}(\mathbf{P}^\top, \mathbf{P}^\top)\mathbf{u}$. Thus, the constraint (4.32) is equivalent to $\mathbf{v}^\top\mathbf{\Lambda}\mathbf{v} = 0$, which indicates $v_1^2 + v_3^2 = v_2^2 + v_4^2$. With this, the optimization problem is simplified as

$$\max_{\lambda,\mathbf{v}}: \quad \lambda \tag{4.34}$$

$$\text{s.t.} \quad v_1^2 + v_3^2 = v_2^2 + v_4^2, \tag{4.35}$$

$$\|\mathbf{v} - \bar{\boldsymbol{\sigma}}\|^2 \leq \eta^2, \tag{4.36}$$

where $\bar{\boldsymbol{\sigma}} = \text{diag}(\mathbf{P}^\top, \mathbf{P}^\top)\boldsymbol{\sigma} = [p_{1,1}\sigma_k,\, p_{1,2}\sigma_k,\, p_{2,1}\sigma_{k+1},\, p_{2,2}\sigma_{k+1}]^\top$. Note that $p_{1,2} = -p_{2,1}$ and $p_{2,2} = p_{1,1}$, we have $\bar{\boldsymbol{\sigma}} = [p_{1,1}\sigma_k,\, -p_{2,1}\sigma_k,\, p_{2,1}\sigma_{k+1},\, p_{1,1}\sigma_{k+1}]^\top$.

Now, problem (4.34) can be solved by checking the feasibility of (4.35) and (4.36) given a particular λ. Given λ, the feasibility of problem(4.34) is equivalent to the feasibility of

$$\min_{v_1^2+v_3^2=v_2^2+v_4^2} \|\mathbf{v} - \bar{\boldsymbol{\sigma}}\|^2 \leq \eta^2. \tag{4.37}$$

Note that $\bar{\boldsymbol{\sigma}}$ depends on λ, we denote the left hand side of inequality (4.37) as $f(\mathbf{v}, \lambda) = \|\mathbf{v} - \bar{\boldsymbol{\sigma}}\|^2$ and parametrize $\mathbf{v}$ as

$$v_1 = r\cos(\alpha), v_2 = r\cos(\beta), v_3 = r\sin(\alpha), v_4 = r\sin(\beta). \tag{4.38}$$

It is easy to verify that the minimum point of $f(\mathbf{v}, \lambda)$ in terms of $\mathbf{v}$ is obtained at the

following stationary point

$$\begin{cases} r = \frac{1}{2}\left(\sqrt{p_{1,1}^2\sigma_k^2 + p_{2,1}^2\sigma_{k+1}^2} + \sqrt{p_{2,1}^2\sigma_k^2 + p_{1,1}^2\sigma_{k+1}^2}\right), \\ \cos(\alpha) = p_{1,1}\sigma_k / \sqrt{p_{1,1}^2\sigma_k^2 + p_{2,1}^2\sigma_{k+1}^2}, \\ \sin(\alpha) = p_{2,1}\sigma_{k+1} / \sqrt{p_{1,1}^2\sigma_k^2 + p_{2,1}^2\sigma_{k+1}^2}, \\ \cos(\beta) = -p_{2,1}\sigma_k / \sqrt{p_{1,1}^2\sigma_{k+1}^2 + p_{2,1}^2\sigma_k^2}, \\ \sin(\beta) = p_{1,1}\sigma_{k+1} / \sqrt{p_{1,1}^2\sigma_{k+1}^2 + p_{2,1}^2\sigma_k^2}. \end{cases} \tag{4.39}$$

Plug the optimal r, α, β of (4.39) into $f(\mathbf{v}, \lambda)$, and we have

$$\begin{aligned} f(\lambda) &\triangleq \min_{v_1^2+v_3^2=v_2^2+v_4^2} f(\mathbf{v}, \lambda) \\ &= (\sigma_k^2 + \sigma_{k+1}^2)/2 \\ &\quad - \sqrt{p_{1,1}^2\sigma_k^2 + p_{2,1}^2\sigma_{k+1}^2}\sqrt{p_{2,1}^2\sigma_k^2 + p_{1,1}^2\sigma_{k+1}^2}. \end{aligned}$$

According to inequality (4.37), inequality $f(\lambda) \leq \eta^2$ now is equivalent to

$$\begin{aligned} &\sqrt{p_{1,1}^2\sigma_k^2 + p_{2,1}^2\sigma_{k+1}^2}\sqrt{p_{2,1}^2\sigma_k^2 + p_{1,1}^2\sigma_{k+1}^2} \\ &\qquad \geq (\sigma_k^2 + \sigma_{k+1}^2)/2 - \eta^2. \end{aligned} \tag{4.40}$$

Denote the right hand of the above inequality as $c \triangleq (\sigma_k^2 + \sigma_{k+1}^2)/2 - \eta^2$. Since $\eta < (\sigma_k - \sigma_{k+1})/\sqrt{2}$, we have $c > \sigma_k\sigma_{k+1}$. Furthermore, we notice that $p_{1,1}^2 = 1 - p_{2,1}^2$. Plug it into inequality (4.40), and we have

$$p_{2,1}^4 - p_{2,1}^2 + \frac{c^2 - \sigma_k^2\sigma_{k+1}^2}{(\sigma_k^2 - \sigma_{k+1}^2)^2} \leq 0. \tag{4.41}$$

Let

$$w \triangleq \frac{c^2 - \sigma_k^2 \sigma_{k+1}^2}{(\sigma_k^2 - \sigma_{k+1}^2)^2}, \tag{4.42}$$

and since $\sigma_k \sigma_{k+1} < c \leq (\sigma_k^2 + \sigma_{k+1}^2)/2$, we have $0 < w \leq \frac{(\sigma_k^2+\sigma_{k+1}^2)^2/4-\sigma_k^2\sigma_{k+1}^2}{(\sigma_k^2-\sigma_{k+1}^2)^2} = 1/4$. Denote the left hand of inequality (4.41) as $h(p_{2,1})$, and we have

$$\begin{aligned} h_{\min} &= h(1/\sqrt{2}) = -1/4 + w \leq 0, \\ h(1) &= w > 0. \end{aligned}$$

Moreover, since $1/\sqrt{2} < p_{2,1} < 1$, we must have

$$p_{2,1} \leq p_{2,1}^H, \tag{4.43}$$

where $p_{2,1}^H = \sqrt{(1+\sqrt{1-4w})/2}$ is the largest root of $h(p_{2,1}) = 0$. Pluging the expressions of $p_{2,1}$ and $p_{2,1}^H$ into (4.43), we can get

$$\frac{\sqrt{\lambda^2+1}+\lambda}{\sqrt{(\sqrt{\lambda^2+1}+\lambda)^2+1}} \leq \sqrt{\frac{1+\sqrt{1-4w}}{2}}.$$

Simplifying this inequality leads to $\lambda \leq \frac{e^2-1}{2e}$, where

$$e = \sqrt{\frac{1+\sqrt{1-4w}}{1-\sqrt{1-4w}}}. \tag{4.44}$$

Thus we can conclude that

$$\lambda_{\max} = \frac{e^2-1}{2e}. \tag{4.45}$$

Accordingly, the optimal subspace distance in (4.1) is

$$\theta^* = \text{atan}(\lambda_{\max})/2. \tag{4.46}$$

In summary, given energy budget η, we first compute w according to (4.42) and compute e according to (4.44), from which we can get $\lambda_{\max}$ and θ^* using (4.45) and (4.46). Having obtained the optimal $\lambda_{\max}$, we can compute $\mathbf{P}$ in (4.33) and compute $\mathbf{v}$ using (4.39) and (4.38), and sequentially compute $\mathbf{u}$ and $\bar{\mathbf{b}}$. Finally, if the optimal solution of problem (4.29) is $\mathbf{B}^*$ with non-zero entries $\bar{\mathbf{b}}^* = [b^*_{k,k}, b^*_{k+1,k}, b^*_{k,k+1}, b^*_{k+1,k+1}]^\top$, we also have another paired feasible optimal solution with non-zero entries being $[b^*_{k,k}, -b^*_{k+1,k}, -b^*_{k,k+1}, b^*_{k+1,k+1}]^\top$, which leads to the same optimal value. Accordingly, the optimal solution to problem (4.1) is $\mathbf{\Delta X}^* = \mathbf{U}\mathbf{B}^*\mathbf{V}^\top$.

4.5 Numerical Experiments and Applications

In this section, we provide numerical examples to illustrate the results obtained in this chapter. We will also apply the results to principal component regression[117] to illustrate potential applications in practice.

4.5.1 Numerical Experiments

In this subsection, we illustrate the results with synthesized data.

In the first experiment, we employ different attack strategies in a low-rank data matrix. In this simulation, we set $d = 5$, $n = 5$, and $k = 3$. We generate the original data matrix as $\mathbf{X} = \mathbf{A}\mathbf{B}^\top$, where $\mathbf{A} \in \mathbb{R}^{d\times k}$, $\mathbf{B} \in \mathbb{R}^{n\times k}$, and each entry of $\mathbf{A}$ and $\mathbf{B}$ is i.i.d. generated according to a standard normal distribution. First, we conduct our optimal rank-one attack strategy. In this strategy, we use the result from the analysis of the optimal rank-one modification matrix to design $\mathbf{a}, \mathbf{b}$ and add the attack matrix $\mathbf{\Delta X} = \mathbf{a}\mathbf{b}^\top$ to the original data matrix $\mathbf{X}$. We then perform SVD on $\hat{\mathbf{X}}$ and select the k leading principal components.

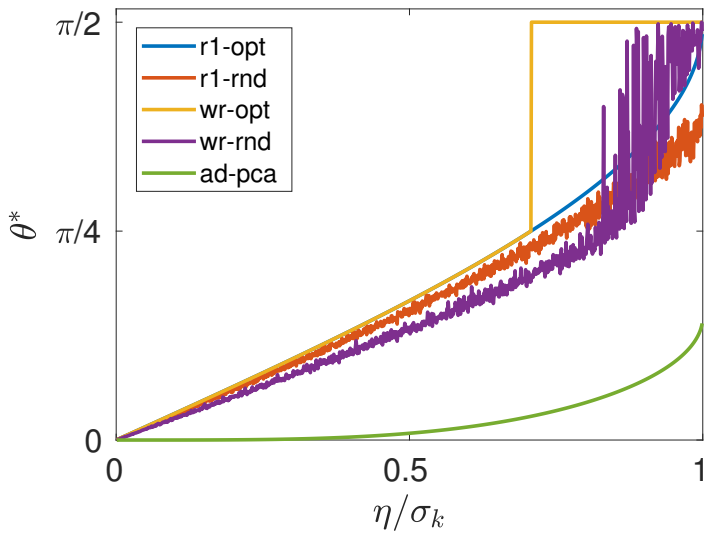


Figure 4.1: Subspace distances with different attack strategies on a low-rank data matrix over different energy budgets.

Finally, we compute the distance between the selected subspace and the original subspace. We also conduct a test using a random rank-one attack strategy, in which we randomly generate $\mathbf{a}, \mathbf{b}$ with each entry of $\mathbf{a}, \mathbf{b}$ being i.i.d. generated according to the standard normal distribution. Then we normalize the energy of $\mathbf{a}\mathbf{b}^\top$to be η^2. For each η, we repeatedly generate 100000 pairs of $\mathbf{a}$ and $\mathbf{b}$ and compute their corresponding subspace distances. In addition, we compare it with the strategy where the modification matrix is free of rank constraint. Although our analysis is deliberately designed for general data matrices, we set the $(k+1)$th singular value to be zero so that it can be applied to the low-rank data matrix. We design the modification matrix $\mathbf{\Delta X}$ according to our analysis in this chapter and calculate the subspace distance between the original subspace and that after modification. Moreover, we conduct another random attack strategy in which we randomly generate the modification matrix without any rank constraint. Each entry of the modification matrix is i.i.d. generated according to a standard normal distribution. After that, we normalize its Frobenius norm equal to η. We repeat this attack 100000 times for each η and record its corresponding subspace distance. Furthermore, we also compare it with the strategy described in [27], which adds one adversarial data sample into the data set.

Fig. 4.1 demonstrates the subspace distances obtained by the five strategies. In this figure, r1-opt represents the rank-one optimal attack obtained in this chapter, r1-rnd represents the maximal subspace distance obtained among the 100000 times random rank-one attacks, wr-opt stands for our optimal attack without the rank constraint, wr-rnd is the maximal subspace distance among the 100000 random attacks without the rank constraint, and ad-pca is the algorithm described in [27]. The x axis is the ratio between η and the smallest singular value of the original data matrix. From the figure, we can see our optimal strategies are much better than the ad-pca strategy. It is because our strategies can modify the data matrix and thus have higher degree of freedom to manipulate the data. The optimal strategies designed in this chapter also have a larger subspace distance compared with their corresponding random attack strategies. In the region where $\eta/\sigma_k \in [0, 1/\sqrt{2}]$, both of our two optimal strategies provide the same subspace distances, which can be verified by setting $\sigma_{k+1} = 0$, computing θ^* in equation (4.46) and comparing it with the value in equation (4.15). When $\eta/\sigma_k > 1/\sqrt{2}$, the optimal attack without the rank constraint leads to the largest subspace distance, $\pi/2$, which is much larger than the distance obtained by the optimal rank-one attack strategy. That means, without the rank constraint, it indeed provides a larger subspace distance.

In the second numerical experiment, we test these strategies except the ad-pca in the general data matrix in which the data matrix is not low-rank. In this experiment, we set $d = 5$, $n = 5$, and $k = 3$. We randomly generate the data matrix $\mathbf{X} \in \mathbb{R}^{d\times n}$ with each entry i.i.d generated according to a standard normal distribution. We also design the optimal rank-one attack matrix and the optimal modification matrix without the rank constraint according to the analysis provided in this chapter. In addition, we do random attacks 100000 times using the randomly generated modification matrix with the rank-one constraint and without the rank constraint, respectively.

Fig. 4.2 shows the subspace distances obtained through different strategies over different energy budgets. In this figure, the x axis is the ratio between η and $\sigma_k - \sigma_{k+1}$. We demonstrate

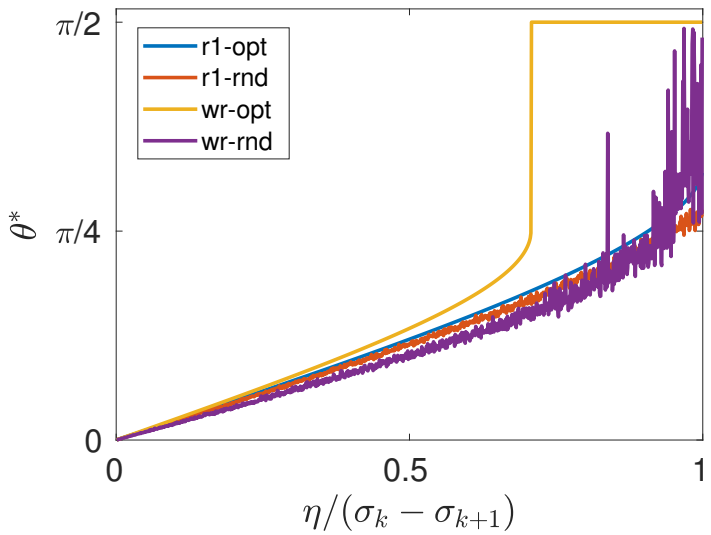


Figure 4.2: Subspace distances achieved by using different attack strategies under different energy budgets.

the maximal subspace distances achieved by the 100000 times random attacks for the two random attack strategies. As the figure shows, both random strategies have smaller subspace distances than their optimal strategies. Unlike, the low-rank case, the strategy without the rank constraint provides larger subspace distances consistently over all the energy budgets.

4.5.2 Applications

In this subsection, we use real data to illustrate the results obtained in this chapter.

In particular, we illustrate the impact of the adversarial attack on PCR, which is widely used in statistical learning, especially when collinearity exists in the data. Ordinary regression will increase the standard error of the coefficients when there are high correlations or even collinearities between features. This happens particularly when the number of features is much larger than the number of data samples. PCR deals with this issue by performing PCA on the feature matrix and only selecting the leading k principal components as the predictors, and thus dramatically decreases the number of predictors. The regression process of PCR can be seen as projecting the response values onto the subspace spanned by the

leading k principal components. So, the accuracy of the subspace will significantly influence the regression results. Appendix G provides an example of how the change of the subspace will influence the result of PCR. More details of PCR can be found in [117].

In this experiment, our task is to use the gasoline spectral intensity to predict its octane rating. We use the gasoline spectral data set [111], which comprises spectral intensities of 60 samples of gasoline at 401 wavelengths and their octane ratings. Fig. 3.4 shows the spectral intensities of the data set. This figure indicates that the correlation of intensity among different wavelengths is very high. To complete the regression task, we can use PCR.

In this experiment, we randomly select 80 percent of the data as the training set and the remaining 20 percent as the test set. We choose 4 principal components as our predictors and perform regression based on these principal components. We also record the r-squared values both in the training phase and the test phase. The r-squared value is defined as $r^2 = 1 - \frac{\|\mathbf{y}-\hat{\mathbf{y}}\|^2}{\|\mathbf{y}-\bar{\mathbf{y}}\|^2}$, where r^2 is the r-squared value, $\mathbf{y}$ is the response values, $\hat{\mathbf{y}}$ is the predicted values, $\|\mathbf{y} - \bar{\mathbf{y}}\|^2$ represents the total variance of the response values, and $\bar{\mathbf{y}} = \text{mean}(\mathbf{y}) \cdot \mathbf{1}$ stands for the mean vector of the response values. R-squared value measures how well the model fits the data and larger r-squared value indicates better regression. Firstly, we perform regular PCR without attack and let na-train and na-test denote the r-squared values of the training and test, respectively. We then attack the feature matrix using the optimal rank-one strategy proposed in this chapter with different energies and denote r1-train and r1-test as its r-squared values in the training and test processes. Finally, we also carry out the optimal attack without the rank constraint and denote wr-train, wr-test as the r-squared values in the training and test procedures.

Fig. 4.3 illustrates the r-squared values with different attack strategies under different energy budgets. As shown in this figure, with the increase of the energy budget, r-squared values of training and test decrease for both attack strategies. This figure also indicates that the strategy with no rank constraint is more efficient than the rank-one strategy considering its smaller r-squared values. Furthermore, the r-squared value of the strategy without the

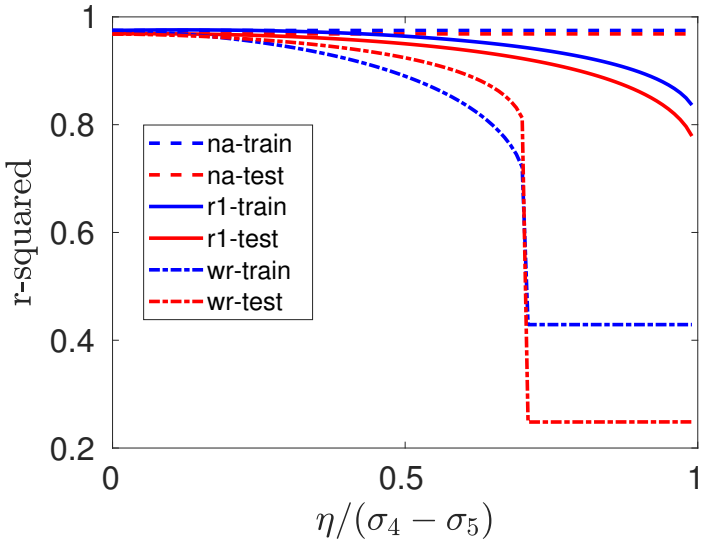


Figure 4.3: R-squared values with different attack strategies over different energy budgets.

rank constraint has a tremendous drop at the point $\eta/(\sigma_4 - \sigma_5) = 1/\sqrt{2}$, which is consistent with our analysis that beyond this particular point, the maximal subspace distance is $\pi/2$.

4.6 Summary

In this chapter, we have investigated the adversarial robustness of the subspace learning problem. We have characterized the optimal rank-one adversarial modification strategy and the optimal strategy without the rank constraint to modify the data. Our analysis has shown that both of the two strategies depend on the singular values of the data matrix and the adversary's energy budget. We have also performed numerical simulations and investigated the impact of this attack on PCR. Both the numerical experiments and the PCR application illustrate that adversarial attacks degrade the performance of subspace learning significantly.

Chapter 5

Conclusions and Extensions

In this chapter, we summarize the contributions of this dissertation and propose several possible extensions.

5.1 Summary

In our dissertation, we have carried out theoretical analyses of the adversarial robustness of some machine learning problems. We stood in the position of the adversary and studied its optimal attack strategy. By investigating its optimal attack strategy, our dissertation gave a clear view of the robustness of the linear regression, LASSO based feature selection, and subspace learning under adversarial attacks.

In Chapter 2, we have investigated how to manipulate the coefficients obtained via linear regression by adding carefully designed poisoning data points to the dataset or modifying the original data points. Given the energy budget, we first provided the closed-form solution of the optimal poisoning data point when our target is modifying one designated regression coefficient. We then extended the analysis to a more challenging scenario where the attacker aims to change one particular regression coefficient while making others to be changed as small as possible. For this scenario, we introduced a semidefinite relaxation method to design the best attack scheme. Finally, we studied a more powerful adversary who can perform

a rank-one modification on the feature matrix. We proposed an alternating optimization method to find the optimal rank-one modification matrix. Numerical examples are provided to illustrate the analytical results obtained in this paper.

In Chapter 3, we have investigated the adversarial robustness of feature selection based on LASSO. In the considered model, a malicious adversary can observe the whole dataset and then carefully modify the response values or the feature matrix to manipulate the selected features. We formulated the modification strategy of the adversary as a bi-level optimization problem. Due to the difficulty of the non-differentiability of the ℓ_1 norm at the zero point, we reformulated the ℓ_1 norm regularizer as linear inequality constraints. We employed the interior-point method to solve this reformulated LASSO problem and obtained the gradient information. Then we used the projected gradient descent method to design the modification strategy. In addition, we demonstrated that this method could be extended to other ℓ_1 based feature selection methods, such as group LASSO and sparse group LASSO. Numerical examples with synthetic and real data illustrated that our method is efficient and effective.

In Chapter 4, we have studied the adversarial robustness of subspace learning problems. Different from the assumptions made in existing works on robust subspace learning where data samples are contaminated by gross sparse outliers or small dense noises, we considered a more powerful adversary who can first observe the data matrix and then intentionally modify the whole data matrix. We first characterized the optimal rank-one attack strategy that maximizes the subspace distance between the subspace learned from the original data matrix and that learned from the modified data matrix. We then generalized the study to the scenario without the rank constraint and characterized the corresponding optimal attack strategy. Besides, our analysis showed that the optimal strategies depend on the singular values of the original data matrix and the adversary's energy budget. Finally, we have provided numerical experiments and practical applications to demonstrate the efficiency of the attack strategies.

5.2 Future Work

One possible extension of this dissertation is to study the defense strategy against our attacks. If we consider the defense strategy, one possible problem formulation is

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \quad \ell_{def}(\hat{\mathbf{X}}, \hat{\mathbf{y}}, \boldsymbol{\beta}) \tag{5.1}$$

$$\text{s.t.} \quad \hat{\mathbf{X}}, \hat{\mathbf{y}} = \underset{\mathbf{X} \in \mathcal{C}_x,\, \mathbf{y} \in \mathcal{C}_y}{\operatorname{argmin}} \quad \ell_{adv}(\mathbf{X}, \mathbf{y}), \tag{5.2}$$

where $\ell_{def}(\cdot)$ is the objective of the defender , $\ell_{adv}(\cdot)$ is the objective of the adversary, and $\mathcal{C}_x$ and $\mathcal{C}_y$ are the modification constraints of the feature matrix and response values, respectively. We should also note that $\ell_{adv}(\cdot)$ may also depend on the defense strategy, which will then render the problem as a competing game between the defender and attacker. With an appropriately designed loss function of the defender, solving this optimization problem leads to the best defense strategy under the optimal attack strategy. The complexity of this problem depends on the forms of $\ell_{def}(\cdot)$, $\ell_{adv}(\cdot)$ and their relationship. In some special cases, we can analyze this problem. For example, Chapter 2 solved this problem when $\ell_{def}(\cdot)$ is the MSE loss function and $\ell_{adv}(\cdot)$ is the objective of manipulating one of the regression coefficients. When $\ell_{def}(\cdot) = -\ell_{adv}(\cdot)$, it is a minmax problem and Jagielski et al. studied this problem when $\ell_{def}(\cdot) = -\ell_{adv}(\cdot)$ and $\ell_{def}(\cdot)$ equals to the MSE loss function [9]. Generally, this problem is very complicated as the upper-level and lower-level optimization problems are interconnected. Hence, how to design $\ell_{def}(\cdot)$ and solve (5.1) efficiently are potential future research topics.

Appendix A

Lasserre's Relaxation Method

In this appendix, we briefly introduce Lasserre's relaxation method and use this method to solve problem (2.37). Lasserre's relaxation method is dedicated to solving the multivariate polynomial optimization problems. A general multivariate polynomial optimization problem contains a multivariate polynomial objective function, $p(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$, and some constraints defined by polynomial inequalities, $g_i(\mathbf{x}) \geq 0$, $i = 1, 2, \ldots, r$:

$$\min : \quad p(\mathbf{x}) \tag{A.1}$$

$$\text{s.t.} \quad g_i(\mathbf{x}) \geq 0,\ i = 1, 2, \ldots, r. \tag{A.2}$$

Clearly, our optimization problem (2.37) can be viewed a multivariate polynomial optimization problem, since in (2.37) the objective function is a fourth order multivariate polynomial and the constraint is a quadratic polynomial.

To proceed, let us explain more details about the problem. The polynomial in the objective, $p(\mathbf{x})$, can be written as:

$$p(\mathbf{x}) = \sum_{\alpha} p_\alpha \mathbf{x}^\alpha, \tag{A.3}$$

where $\alpha \in \mathbb{N}^n$,

$$\mathbf{x}^\alpha = \prod_{i=1}^{n} x_i^{\alpha^i}, \tag{A.4}$$

and $|\alpha| = \sum_i \alpha^i$. Suppose the order of the objective function is m_0, we have $|\alpha| \leq m_0$. Define $\mathbf{p}_\alpha = \{p_\alpha\} \in \mathbb{R}^{s(m_0)}$ as the coefficients of the polynomial basis $\{1, x_1, x_2, \ldots, x_n, x_1^2, x_1x_2, \ldots, x_n^{m_0}\}$. Hence, the dimension of the basis is $s(m_0) = \binom{n+m_0}{m_0}$. Instead of directly solving problem (A.1), Lasserre's relaxation method [83] first converts it into the following equivalent problem

$$\min_{\mu \in \mathcal{P}(\mathcal{K})} : \quad \int p(\mathbf{x})\, \mathrm{d}(\mu(\mathbf{x})), \tag{A.5}$$

where $\mathcal{K}$ is the semialgebraic set defined by the inequalities: $\mathcal{K} = \{\mathbf{x} \,|\, g_i(\mathbf{x}) \geq 0, i = 1, 2, \ldots, r\}$, and $\mathcal{P}(\mathcal{K})$ is the set of all probability measures supported on $\mathcal{K}$.

To see that problem (A.1) and (A.5) are equivalent, suppose the optimal values of (A.1) and (A.5) are p_0^* and p^*, respectively. Since $p(\mathbf{x}) \geq p_0^*$, we have $p^* \geq p_0^*$. Conversely, suppose the optimal solution of (A.1) is x^*, $\mu = \delta_{x^*}$ is a feasible solution to (A.5). Hence, we also have $p^* \leq p_0^*$. Thus, the two problems are equivalent.

With the help of this reformulation, finding the global optimal points for (A.1) is equivalent to finding the optimal distribution of (A.5). Since $\int p(\mathbf{x})\, \mathrm{d}\mu(\mathbf{x}) = \sum_\alpha p_\alpha \int \mathbf{x}^\alpha\, \mathrm{d}\mu(\mathbf{x})$, the objective function of (A.5) is just $\mathbf{p}_\alpha^\top \mathbf{y}_\alpha$, where $\mathbf{y}_\alpha = \{y_\alpha\}$ and $y_\alpha = \int \mathbf{x}^\alpha\, \mathrm{d}\mu(\mathbf{x})$. So, finding the optimal probability is identical to finding the optimal $\mathbf{y}_\alpha$ under the constraint that $\mathbf{y}_\alpha$ is a valid moment sequence with respect to some probability measure on $\mathcal{K}$. The solution to this problem is fully characterized by the K-moment problem in case $\mathcal{K}$ is compact. Let us give more notations for the convenience of introducing this method.

Given an $s(2m)$ length vector, $\mathbf{y}_\alpha = \{y_\alpha\}$, with its first element $y_{0,\ldots,0} = 1$. The $s(m)$ dimensional moment matrix $M_m(y)$ is constructed as follows: the first row and columns is defined as $M_m(1, k) = y_{\alpha_k}$ and $M_m(k, 1) = y_{\alpha_k}$ for $k = 1, 2, \ldots, s(m)$ and $M_m(i, j) = y_{\alpha_i + \alpha_j}$

for $i, j = 2, \ldots, s(m)$. For instance, when $n = 2, m = 2$,

$$M_m(y) = \begin{bmatrix} 1 & y_{10} & y_{01} & y_{20} & y_{11} & y_{02} \\ y_{10} & y_{20} & y_{11} & y_{30} & y_{21} & y_{12} \\ y_{01} & y_{11} & y_{02} & y_{21} & y_{12} & y_{03} \\ y_{20} & y_{30} & y_{21} & y_{40} & y_{31} & y_{22} \\ y_{11} & y_{21} & y_{12} & y_{31} & y_{22} & y_{13} \\ y_{02} & y_{12} & y_{03} & y_{22} & y_{13} & y_{04} \end{bmatrix}.$$

Moreover, $M_m(y)$ defines a bi-linear form, $\langle \cdot, \cdot \rangle$, on two polynomials

$$\langle p, q \rangle_y = \langle p, M_m(y) q \rangle = \sum_\alpha (pq)_\alpha y_\alpha = \int p(\mathbf{x}) q(\mathbf{x}) \, \mathrm{d}\mu(\mathbf{x}).$$

So, if $\mathbf{y}_\alpha$ is a sequence of moments of some probability measure, we have

$$\langle q, q \rangle_y = \int q(x)^2 \, \mathrm{d}(\mu(\mathbf{x})) \geq 0.$$

Thus, we have $M_m(y) \succcurlyeq 0$. Let $p(\mathbf{x})$ be a multivariate polynomial with coefficient vector $\mathbf{p}_\beta = \{p_\beta\}$, and define the localizing matrix $M_m(py)$ as

$$M_m(py)(i, j) = \sum_\beta p_\beta y_{\alpha_i + \alpha_j + \beta}.$$

For example, with

$$M_1(y) = \begin{bmatrix} 1 & y_{10} & y_{01} \\ y_{10} & y_{20} & y_{11} \\ y_{01} & y_{11} & y_{02} \end{bmatrix} \quad \text{and} \quad p(\mathbf{x}) = a - x_1^2 - x_2^2,$$

we have

$$M_1(py) = \begin{bmatrix} a - y_{20} - y_{02} & ay_{10} - y_{30} - y_{12} & ay_{01} - y_{21} - y_{03} \\ ay_{10} - y_{30} - y_{12} & ay_{20} - y_{40} - y_{22} & ay_{11} - y_{31} - y_{13} \\ ay_{01} - y_{21} - y_{03} & ay_{11} - y_{31} - y_{13} & ay_{01} - y_{22} - y_{04} \end{bmatrix}.$$

Also, if $p(\mathbf{x}) \geq 0$, by definition, we have $M_m(py) \succcurlyeq 0$.

Further, we make the following assumption on the semialgebraic set $\mathcal{K}$.

Assumption 1. *The set $\mathcal{K}$ is compact and there exists a real-valued polynomial $u(\mathbf{x})$: $\mathbb{R}^n \to \mathbb{R}$ such that $\{u(\mathbf{x}) \geq 0\}$ is compact and*

$$u(\mathbf{x}) = u_0(\mathbf{x}) + \sum_{k=1}^{r} g_i(\mathbf{x}) u_i(\mathbf{x}) \ \textit{for all} \ \mathbf{x} \in \mathbb{R}^n, \tag{A.6}$$

where the polynomial $u_i(\mathbf{x})$ is the sum of squares for $i = 0, 1, \ldots, r$.

Assumption 1 is satisfied in many cases. For example, this assumption is satisfied when there is only one inequality constraint that is compact, which is the case in our problem (2.37).

With the help of the notations and Assumption 1, we have the main result. Let $w_i = \lceil m_i/2 \rceil$, where $m_i, i = 1, 2, \ldots, r$, is the order of $g_i(\mathbf{x})$ and m_0 is the order of the objective, with $N \geq \max\{w_i\}$ for $i = 0, 1, \ldots, r$. Consider the following semidefinite programming

$$\begin{aligned} \min: \quad & \sum_{\alpha} p_\alpha y_\alpha \\ \text{s.t.} \quad & M_N(y) \succcurlyeq 0, \\ & M_{N-w_i}(g_i y) \succcurlyeq 0, \ i = 1, 2, \ldots, r, \end{aligned} \tag{A.7}$$

where N is called the relaxation order. Lasserre [83] shows that as N approaches infinity, the solution of (A.7) converges to the solution of (A.5). However, the dimension of the semidefinite programming (A.7) grows rapidly as N increases and infinite N makes solving

problem (A.7) infeasible. Fortunately, in practice, a small N is enough to get a very good approximation of problem (A.5) [83]. Furthermore, a small N is usually sufficient to get the global optimal solutions and the sufficient rank condition, $\text{rank}M_N(y) = \text{rank}M_{N-w_{\max}}(y)$, where $w_{\max} = \max\{w_i\}, i = 0, 1, \ldots, r$, assures the global optimality. Therefore, after we solving problem (A.7) we are ready to check whether we reach the global optimality. Besides, Henrion and Lasserre developed a systematic way to extract all the optimal solutions in case the rank condition is satisfied [118]. Since our problem (2.37) is just a special case of multivariate polynomial optimization, with the help of this relaxation method, we can solve problem (2.37).

Appendix B

Poof of the Equivalence of Problem (4.9) and Problem (4.10)

Before giving the proof, we first examine the unitary invariant property of the Asimov distance, which is helpful in our subsequent proof.

Proposition B.1. *Let* $\mathbf{P}$ *and* $\mathbf{T}$ *be unitary matrices, and then for the Asimov distance function* $\theta(\cdot,\cdot)$*, we have*

$$\theta\big(\mathbf{X}_1, g_k(\mathbf{X}_2)\big) = \theta\big(\mathbf{P}\mathbf{X}_1\mathbf{T}^\top, g_k(\mathbf{P}\mathbf{X}_2\mathbf{T}^\top)\big).$$

Proof. First, we show $\theta(\mathbf{X}_1, \mathbf{X}_2) = \theta(\mathbf{P}\mathbf{X}_1\mathbf{T}^\top, \mathbf{P}\mathbf{X}_2\mathbf{T}^\top)$. Suppose the thin QR decompositions of $\mathbf{X}_1$ and $\mathbf{X}_2$ are $\mathbf{X}_1 = \mathbf{Q}_1\mathbf{R}_1$, $\mathbf{X}_2 = \mathbf{Q}_2\mathbf{R}_2$, and then the subspace distance between the two subspaces spanned by the columns of $\mathbf{X}_1$ and $\mathbf{X}_2$ is determined by the singular values of $\mathbf{Q}_1^\top\mathbf{Q}_2$. Since $(\mathbf{P}\mathbf{Q}_1)^\top(\mathbf{P}\mathbf{Q}_2) = \mathbf{Q}_1^\top\mathbf{Q}_2$ and right multiplying an unitary matrix does not change the singular values and the column subspace of a matrix, we have $\theta(\mathbf{X}_1, \mathbf{X}_2) = \theta(\mathbf{P}\mathbf{X}_1\mathbf{T}^\top, \mathbf{P}\mathbf{X}_2\mathbf{T}^\top)$.

Second, suppose the full SVD of $\mathbf{X}_2$ is $\mathbf{X}_2 = \mathbf{U}_2\mathbf{\Sigma}_2\mathbf{V}_2^\top$, where $\mathbf{U}_2 = [\mathbf{u}_{21}, \mathbf{u}_{22}, \cdots, \mathbf{u}_{2d}]$.

Then

$$\mathbf{P}g_k(\mathbf{X}_2) = \mathbf{P}[\mathbf{u}_{21}, \mathbf{u}_{22}, \cdots, \mathbf{u}_{2k}] = g_k(\mathbf{P}\mathbf{X}_2),$$

which can be verified by checking that $\mathbf{P}\mathbf{U}_2\mathbf{\Sigma}_2\mathbf{V}_2^\top$ is a valid SVD of $\mathbf{P}\mathbf{X}_2$. It completes the proof. □

With the help of this proposition, let $\mathbf{P} = \mathbf{U}^\top$, $\mathbf{T} = \mathbf{V}^\top$, right multiply $\mathbf{P}$ and left multiply $\mathbf{T}^\top$ on both $\mathbf{X}$ and $\hat{\mathbf{X}}$, and we can simplify problem (4.9) as the following

$$\begin{aligned} \max_{\mathbf{a}\in\mathbb{R}^d, \mathbf{b}\in\mathbb{R}^n} : \quad & \theta(\mathbf{\Sigma}, g_k(\tilde{\mathbf{Y}})) \\ \text{s.t.} \quad & \tilde{\mathbf{Y}} = \mathbf{\Sigma} + \mathbf{a}\mathbf{b}^\top, \\ & \|\mathbf{a}\|\|\mathbf{b}\| \leq \eta, \end{aligned} \tag{B.1}$$

where we assume $n > d$, $\mathbf{\Sigma} = [\text{diag}(\sigma_1, \sigma_2, \cdots, \sigma_k, \mathbf{0}), \mathbf{0}] \in \mathbb{R}^{d\times n}$. Also, from problem (4.9) to problem (B.1), we do variable change $\mathbf{a} \Leftarrow \mathbf{U}^\top\mathbf{a}, \mathbf{b} \Leftarrow \mathbf{V}^\top\mathbf{b}$.

To further simplify this optimization problem, we split $\mathbf{a}$ and $\mathbf{b}$ into $\mathbf{a} = [\mathbf{a}_1^\top, \mathbf{a}_2^\top]^\top, \mathbf{b} = [\mathbf{b}_1^\top, \mathbf{b}_2^\top]^\top$, where $\mathbf{a}_1 \in \mathbb{R}^k$, $\mathbf{a}_2 \in \mathbb{R}^{d-k}$, $\mathbf{b}_1 \in \mathbb{R}^k$, and $\mathbf{b}_2 \in \mathbb{R}^{n-k}$. In addition, utilizing the Householder transformation[78], we construct an orthogonal matrix

$$\mathbf{M}_1 = \begin{bmatrix} \mathbf{I}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_1 \end{bmatrix}, \tag{B.2}$$

where

$$\begin{aligned} &\mathbf{M}_1^\top\mathbf{M}_1 = \mathbf{I}, && \mathbf{H}_1 = \mathbf{I} - 2\frac{\mathbf{u}\mathbf{u}^\top}{\|\mathbf{u}\|^2}, \\ &\mathbf{u} = \mathbf{a}_2 - s_1\|\mathbf{a}_2\| \cdot \mathbf{e}_1, && \mathbf{e}_1 = [1, 0, \cdots, 0]^\top \in \mathbb{R}^{d-k}, \\ &\mathbf{H}_1^\top\mathbf{a}_2 = s_1\|\mathbf{a}_2\| \cdot \mathbf{e}_1, && s_1 = \pm 1. \end{aligned}$$

Similarly, we can construct another Householder transformation matrix $\mathbf{H}_2$ for $\mathbf{b}_2$ and the

corresponding orthogonal matrix $\mathbf{M}_2 = \mathrm{diag}(\mathbf{I}_k, \mathbf{H}_2)$. Left multiplying $\mathbf{M}_1^\top$ and right multiplying $\mathbf{M}_2$ on $\tilde{\mathbf{Y}}$, we have

$$\mathbf{M}_1^\top \tilde{\mathbf{Y}} \mathbf{M}_2 = \begin{bmatrix} \tilde{\mathbf{\Sigma}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{a}_1 \\ s_1 \|\mathbf{a}_2\| \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{b}_1^\top & s_2 \|\mathbf{b}_2\| & \mathbf{0} \end{bmatrix},$$

where $s_2 = \pm 1$.

Let $\mathbf{a} \triangleq [\mathbf{a}_1^\top, s_1\|\mathbf{a}_2\|]^\top$ and $\mathbf{b} \triangleq [\mathbf{b}_1^\top, s_2\|\mathbf{b}_2\|]^\top$. Utilizing Proposition B.1, it is clear that problem (4.10) and problem (B.1) are equivalent.

Appendix C

Proof of Theorem 4.1

The proof follows similar steps to those in [27]. In problem (4.10), $\tilde{\mathbf{\Sigma}}$ is a diagonal matrix with diagonal elements $\{\sigma_1, \sigma_2, \cdots, \sigma_k, 0\}$. The subspace spanned by $g_k(\mathbf{Y})$ is a k-dimensional subspace in $\mathbb{R}^{k+1}$. We denote this subspace as $\mathbb{Q}$, denote $\mathbb{P}$ as the subspace spanned by $\tilde{\mathbf{\Sigma}}$ and further denote their intersection as $\mathbb{T} = \mathbb{P} \cap \mathbb{Q}$. Note that $\mathbb{P}$ is not equal to $\mathbb{Q}$ (otherwise the Asimov distance will be zero), so we have $\dim(\mathbb{P} \cup \mathbb{Q}) = k+1$. Since $\dim(\mathbb{P}) + \dim(\mathbb{Q}) - \dim(\mathbb{T}) = \dim(\mathbb{P} \cup \mathbb{Q})$, we have $\dim(\mathbb{T}) = k-1$. Let $\mathbf{T}$ be an orthonormal basis of $\mathbb{T}$. Let $[\mathbf{T}, \mathbf{p}]$ be an orthonormal basis of $\mathbb{P}$ and let $[\mathbf{T}, \mathbf{q}]$ be an orthonormal basis of $\mathbb{Q}$. By the definition of Asimov distance, the subspace distance between $\mathbb{P}$ and $\mathbb{Q}$ is the angle between $\mathbf{p}$ and $\mathbf{q}$.

Firstly, it is easy to see that $a_{k+1} \neq 0$. Otherwise, $\mathbb{Q}$ will be equal to $\mathbb{P}$, which means that their Asimov distance is zero.

Secondly, it is easy to see $\mathbf{q} \in \text{span}[\mathbf{T}, \mathbf{p}, \mathbf{e}_{k+1}]$, where $\mathbf{e}_{k+1}$ is an ordinary basis vector that only has element 1 in the $(k+1)$th coordinate. Since $\mathbf{T}$ is orthogonal to $\mathbf{q}$, we have $\mathbf{q} \in \text{span}[\mathbf{p}, \mathbf{e}_{k+1}]$. It is easy to see that the larger variance in the direction of $\mathbf{p}$ is, the closer $\mathbf{p}$ and $\mathbf{q}$ will be. Then we should select $\mathbf{p}$ as the direction with the smallest variance in $\mathbf{X}$. Since we are assuming that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k$, $\mathbf{p}$ should be $\mathbf{e}_k$.

Thirdly, for a fixed direction of $\mathbf{a}$, let $\hat{\mathbf{a}}$ be the projection of $\mathbf{a}$ onto $\text{span}[\mathbf{e}_k, \mathbf{e}_{k+1}]$. Clearly,

$\mathbf{q}$ will be closer to $\mathbf{a}$ as $\hat{\mathbf{a}}$ grows. As a result, the angle between $\mathbf{q}$ and $\mathbf{p}$ will be larger. This also implies that the length of $\mathbf{a}$ should be maximized: $\|\mathbf{a}\| = \eta$. Hence, the Asimov distance is maximized when $\mathbf{a} = \hat{\mathbf{a}}$ and $\|\mathbf{a}\| = \eta$, implying that $\mathbf{a}$ only has nonzero elements in its kth and $k+1$th coordinates.

Finally, for a fixed $\mathbf{a}$ in the form of (4.11), the projected variance of $\mathbf{Y}$ on the direction of $\mathbf{e}_k$ is $v_1 = \sum_{i \neq k}(a_k b_i)^2 + (a_k b_k + \sigma_k)^2 = a_k^2 + \sigma_k^2 + 2a_k b_k \sigma_k$ and the projected variance of $\mathbf{Y}$ on the direction of $\mathbf{e}_{k+1}$ is $v_2 = \sum_i (a_{k+1} b_i)^2 = a_{k+1}^2$. To maximize the Asimov distance, we need to make v_1 small and v_2 large. Apparently, for fixed $\mathbf{a}$, v_1 is minimized when $b_k = -\text{sign}(a_k)$, which implies $b_i = 0, \forall i \neq k$. To avoid the sign ambiguity, we set $b_k = 1$.

Appendix D

Proof of Theorem 4.2

This proof follows similar steps in the proof of the low-rank case. Denote $\mathbb{P}$ as the subspace spanned by $g_k(\mathbf{\Sigma})$ and $\mathbb{Q}$ as the subspace spanned by $g_k(\mathbf{Y})$, and denote their intersection as $\mathbb{T} = \mathbb{P} \cap \mathbb{Q}$. We further denote $\mathbf{T}$ as an orthonormal basis of $\mathbb{T}$, $[\mathbf{T}, \mathbf{p}]$ as an orthonormal basis of $\mathbb{P}$, and $[\mathbf{T}, \mathbf{q}]$ as an orthonormal basis of $\mathbb{Q}$. From the definition of Asimov distance, the subspace distance between $\mathbb{P}$ and $\mathbb{Q}$ is the subspace distance between the span of $\mathbf{p}$ and the span of $\mathbf{q}$.

First, it is apparent that $\mathbf{q} \in \text{span}[\mathbf{T}, \mathbf{p}, \mathbf{e}_{k+1}, \mathbf{e}_{k+2}, \cdots, \mathbf{e}_d]$. Since $\mathbf{q} \perp \mathbf{T}$, we have $\mathbf{q} \in \text{span}[\mathbf{p}, \mathbf{e}]$, where $\mathbf{e} \in \text{span}[\mathbf{e}_{k+1}, \cdots, \mathbf{e}_d]$. It is easy to see that the subspace distance between the span of $\mathbf{q}$ and the span of $\mathbf{p}$ will be large if the variance of $\mathbf{\Sigma}$ in the span of $\mathbf{p}$ is large and the variance of $\mathbf{\Sigma}$ in the span of $\mathbf{q}$ is small. So we should select $\mathbf{p}$ as the direction in $\text{span}[\mathbf{e}_1, \cdots, \mathbf{e}_k]$ that has the smallest variance of $\mathbf{\Sigma}$ and select $\mathbf{e}$ as the direction among $\text{span}[\mathbf{e}_{k+1}, \cdots, \mathbf{e}_n]$ that has the largest variance of $\mathbf{\Sigma}$. Since $\mathbf{e} \in \text{span}[\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_d]$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k \geq \sigma_{k+1} \geq \cdots \geq \sigma_d$, $\mathbf{p}$ should be $\mathbf{e}_k$ and $\mathbf{e}$ should be $\mathbf{e}_{k+1}$. So, we have $\mathbf{q} \in \text{span}[\mathbf{e}_k, \mathbf{e}_{k+1}]$.

Second, for a fixed direction of $\mathbf{a}$, let $\hat{\mathbf{a}}$ be the projection of $\mathbf{a}$ onto $\text{span}[\mathbf{e}_k, \mathbf{e}_{k+1}]$. It is easy to see that $\mathbf{q}$ will be closer to $\mathbf{a}$ as $\hat{\mathbf{a}}$ grows, and as a result, the angle between $\mathbf{q}$ and $\mathbf{p}$ will be larger. This implies the length of $\mathbf{a}$ should be maximized, which indicates $\|\mathbf{a}\| = \eta$

and the distance is maximized when $\mathbf{a} = \hat{\mathbf{a}}$. It also indicates $a_i = 0$ if $i \neq k$, $k+1$.

Finally, for a fixed $\mathbf{a}$ in the form of (4.18), the projected variance of $\mathbf{Y}$ in the direction of $\mathbf{e}_k$ is $v_k = \sum_{i \neq k}(a_k b_i)^2 + (a_k b_k + \sigma_k)^2 = a_k^2 + \sigma_k^2 + 2a_k b_k \sigma_k$ and the projected variance of $\mathbf{Y}$ in the direction of $\mathbf{e}_{k+1}$ is $v_{k+1} = \sum_{i \neq k+1}(a_{k+1} b_i)^2 + (\sigma_{k+1} + a_{k+1} b_{k+1})^2 = a_{k+1}^2 + \sigma_{k+1}^2 + 2a_{k+1} b_{k+1}$. To maximize the Asimov distance, we should make v_k small and make v_{k+1} large. With the constraint that $\|\mathbf{b}\| = 1$, we should have $b_k^2 + b_{k+1}^2 = 1$, which implies $b_i = 0$ for all $i \neq k$ and $i \neq (k+1)$.

As shown above, the optimal $\mathbf{a}$ and $\mathbf{b}$ should be in the form of (4.18) and (4.19), which completes our proof.

Appendix E

Proof of Theorem 4.3

The optimal solution to problem (4.25) either locates at the boundary or the stationary points.

We first characterize the stationary points. At the stationary points, the value (α^*, β^*) satisfies the necessary conditions

$$\begin{cases} \frac{\partial}{\partial \alpha} |\cos \varphi(\alpha, \beta)|_{\alpha=\alpha^*, \beta=\beta^*} = 0, \\ \frac{\partial}{\partial \beta} |\cos \varphi(\alpha, \beta)|_{\alpha=\alpha^*, \beta=\beta^*} = 0. \end{cases} \tag{E.1}$$

Since $\sin \varphi^* \neq 0$, we have

$$\begin{cases} \frac{\partial}{\partial \alpha} \varphi(\alpha, \beta)|_{\alpha=\alpha^*, \beta=\beta^*} = 0, \\ \frac{\partial}{\partial \beta} \varphi(\alpha, \beta)|_{\alpha=\alpha^*, \beta=\beta^*} = 0, \end{cases} \tag{E.2}$$

in which

$$\begin{aligned}\frac{\partial\varphi}{\partial\alpha} =& \frac{\eta}{a_x^2+a_y^2}\Big(\eta(3\sigma_{k+1}^2-\sigma_k^2+\eta^2)+2\eta(\sigma_k^2-\sigma_{k+1}^2)\cos^2(\alpha)+2\eta(\sigma_k^2-\sigma_{k+1}^2)\cos^2(\beta)\\ &+\sigma_k(\sigma_k^2-\sigma_{k+1}^2+3\eta^2)\cos(\alpha)\cos(\beta)+\sigma_{k+1}(\sigma_{k+1}^2-\sigma_k^2+3\eta^2)\sin(\alpha)\sin(\beta)\Big),\\ \frac{\partial\varphi}{\partial\beta} =& \frac{\eta}{a_x^2+a_y^2}\Big(\sigma_k(\sigma_{k+1}^2+\eta^2-\sigma_k^2)\sin(\alpha)\sin(\beta)\\ &+\sigma_{k+1}(\sigma_k^2+\eta^2-\sigma_{k+1}^2)\cos(\alpha)\cos(\beta)+2\eta\sigma_k\sigma_{k+1}\Big).\end{aligned}$$

Eliminating $\sin(\alpha)\sin(\beta)$ from (E.2), we have

$$C\cos^2(\alpha)+D\cos(\alpha)\cos(\beta)+C\cos^2(\beta)+F=0, \tag{E.3}$$

where

$$C=2\eta\sigma_k(\sigma_k^2-\sigma_{k+1}^2)(\sigma_{k+1}^2+\eta^2-\sigma_k^2),$$

$$D=(\sigma_k^2-\sigma_{k+1}^2)\left(-(\sigma_k^2-\sigma_{k+1}^2)^2-2\eta^2(\sigma_k^2+\sigma_{k+1}^2)+3\eta^4\right),$$

and

$$F=\eta\sigma_k\left(\sigma_k^4+\sigma_{k+1}^4+\eta^4-2\sigma_k^2\sigma_{k+1}^2-2\sigma_k^2\eta^2-2\sigma_{k+1}^2\eta^2\right).$$

Further, we rewrite the first equation of (E.2) as

$$c\sqrt{(1-\cos^2(\alpha))(1-\cos^2(\beta))}+d\cos(\alpha)\cos(\beta)+e=0, \tag{E.4}$$

where $c=\sigma_k(\sigma_{k+1}^2+\eta^2-\sigma_k^2)$, $d=\sigma_{k+1}(\sigma_k^2+\eta^2-\sigma_{k+1}^2)$, and $e=2\eta\sigma_k\sigma_{k+1}$.

Combining (E.3) and (E.4) and eliminating $\cos^2(\alpha)$ and $\cos^2(\beta)$, we have

$$(c^2-d^2)\cos(\alpha)^2\cos(\beta)^2+\left(\frac{Dc^2}{C}-2de\right)\cos(\alpha)\cos(\beta)+\frac{c^2F}{C}+c^2-e^2=0.$$

The left side of the equation is a quadratic function with respect to $r=\cos(\alpha)\cos(\beta)$. The

two roots are:

$$r_1 = -\frac{\sigma_k \eta}{\sigma_k^2 - \sigma_{k+1}^2}, \quad r_2 = -\frac{\sigma_k}{2}\left(\frac{1}{\eta} + \frac{\eta}{\sigma_k^2 - \sigma_{k+1}^2}\right).$$

Note that $\eta \in [0, \sigma_k - \sigma_{k+1})$, so we have $r_1 \in (-\frac{\sigma_k}{\sigma_k+\sigma_{k+1}}, 0]$, $r_2 \in (-\infty, -\frac{\sigma_k}{\sigma_k-\sigma_{k+1}})$. Since $|\cos(\alpha)\cos(\beta)| \leq 1$, $\frac{\sigma_k}{\sigma_k+\sigma_{k+1}} < 1$, and $\frac{\sigma_k}{\sigma_k-\sigma_{k+1}} > 1$, we should only retain the first root r_1. Substitute $\cos(\alpha)\cos(\beta) = r_1 = -\frac{\eta\sigma_k}{\sigma_k^2-\sigma_{k+1}^2}$ into (E.3), and we have $C\cos^4(\alpha) + (Dr_1 + F)\cos^2(\alpha) + Cr_1^2 = 0$. The left side of the equation is a quadratic function with respect to $s = \cos^2(\alpha)$, so we can easily find its roots. Let us denote s_1 and s_2 as the two roots:

$$s_1 = \frac{\sigma_k^2 - \sigma_{k+1}^2 + \eta^2 - \sqrt{H}}{2(\sigma_k^2 - \sigma_{k+1}^2)}, \quad s_2 = \frac{\sigma_k^2 - \sigma_{k+1}^2 + \eta^2 + \sqrt{H}}{2(\sigma_k^2 - \sigma_{k+1}^2)},$$

where $H = \sigma_k^4 + \sigma_{k+1}^4 + \eta^4 - 2\sigma_k^2\sigma_{k+1}^2 - 2\sigma_k^2\eta^2 - 2\sigma_{k+1}^2\eta^2$. We need to check that H is positive. Viewing H as a function of η and taking derivative, we have $H'(\eta) = 2\eta(2\eta^2 - 2(\sigma_k^2+\sigma_{k+1}^2)) < 0$. Since $\eta^2 \in [0, (\sigma_k - \sigma_{k+1})^2)$, we have $H(\eta) \in (0, (\sigma_k^2 - \sigma_{k+1}^2)^2]$.

As $\cos(\alpha)^2 \leq 1$, we need to check whether $s_1, s_2 \in [0, 1]$.

Firstly, as H is a decreasing function of η in the considered range, s_1 is a increasing function of η. Therefore, we have $\min(s_1) = s_1(\eta)|_{\eta=0} = 0$ and $\max(s_1) = s_1(\eta)|_{\eta=\sigma_k-\sigma_{k+1}} = \frac{\sigma_k}{\sigma_k+\sigma_{k+1}} < 1$. Hence, s_1 is a valid solution.

Secondly, it is easy to check that s_2 is a decreasing function of η. So, we have $\max(s_2) = s_2(\eta)|_{\eta=0} = 1$ and $\min(s_2) = s_2(\eta)|_{\eta=\sigma_k-\sigma_{k+1}} = \frac{\sigma_k}{\sigma_k+\sigma_{k+1}} < 1$, which means s_2 is also a valid solution. Hence, we have two stationary points

$$\begin{cases} \cos^2(\alpha) & = \frac{\sigma_k^2-\sigma_{k+1}^2+\eta^2 \pm \sqrt{H}}{2(\sigma_k^2-\sigma_{k+1}^2)}, \\ \cos^2(\beta) & = \frac{\sigma_k^2-\sigma_{k+1}^2+\eta^2 \mp \sqrt{H}}{2(\sigma_k^2-\sigma_{k+1}^2)}. \end{cases} \tag{E.5}$$

Since there are two sets of solutions in (E.5), we should determine which one is better. The variance of $\mathbf{Y}$ in the direction of $\mathbf{e}_k$ is $v_k = \cos^2(\alpha) + \sigma_k^2 + 2\cos(\alpha)\cos(\beta)$ and the variance

of $\mathbf{Y}$ in the direction of $\mathbf{e}_{k+1}$ is $v_{k+1} = \sin^2(\alpha) + \sigma_{k+1}^2 + 2\sin(\alpha)\sin(\beta)$. Both of the two sets of solutions in (E.5) lead to $\cos(\alpha)\cos(\beta) = -\frac{\eta\sigma_k}{\sigma_k^2 - \sigma_{k+1}^2}$ and $\sin(\alpha)\sin(\beta) = \frac{\eta\sigma_{k+1}}{\sigma_k^2 - \sigma_{k+1}^2}$. For fixed $\cos(\alpha)\cos(\beta)$ and $\sin(\alpha)\sin(\beta)$, the smaller $\cos^2(\alpha)$ is, the smaller v_k will be, and the larger the subspace distance will be. Hence, we conclude the stationary point that satisfies

$$\begin{cases} \cos^2(\alpha^*) & = \frac{\sigma_k^2 - \sigma_{k+1}^2 + \eta^2 - \sqrt{H}}{2(\sigma_k^2 - \sigma_{k+1}^2)} \\ \cos^2(\beta^*) & = \frac{\sigma_k^2 - \sigma_{k+1}^2 + \eta^2 + \sqrt{H}}{2(\sigma_k^2 - \sigma_{k+1}^2)} \end{cases} \tag{E.6}$$

leads to a larger subspace distance.

Finally, it is easy to compute the objective values of problem (4.25) at the boundary points. Comparing these values with the objective values induced by the point in equation (E.6), we can readily conclude the point in equation (E.6) gives a larger objective value. In summary, given that $\alpha \in [0, \pi/2]$ and $\beta \in [\pi/2, \pi]$, the optimal α and β are shown in (4.26).

Appendix F

Proof of Theorem 4.4

The proof has two main steps. In the first step, we show that non-zero entries of $\mathbf{B}$ are in the kth and $(k+1)$th rows. In the second step, we will further prove the entries except in the kth and $(k+1)$th columns should be zero.

In the first step, we follow similar proof procedures in Theorem 4.2. We use $\mathbb{P}$ to denote the subspace spanned by $g_k(\mathbf{\Sigma})$ and $\mathbb{Q}$ to denote the subspace spanned by $g_k(\mathbf{Y})$. We also use $\mathbb{T}$ to represent the intersection of the two subspaces and further denote $\mathbf{T}$ as one set of orthonormal bases of $\mathbb{T}$, $[\mathbf{T}, \mathbf{p}]$ as one set of orthonormal bases of $\mathbb{P}$ and $[\mathbf{T}, \mathbf{q}]$ as one set of orthonormal bases of $\mathbb{Q}$. So, the subspace distance between $\mathbb{P}$ and $\mathbb{Q}$ is the subspace distance between the subspace spanned by $\mathbf{p}$ and that spanned by $\mathbf{q}$. Following the same arguments in Theorem 4.2, by setting all the entries of $\mathbf{B}$ to be zero except the kth and $(k+1)$th rows, we can guarantee achieving the maximal subspace distance and further we have $\mathbf{q} \in \text{span}[\mathbf{e}_k, \mathbf{e}_{k+1}]$ and $\mathbf{p} = \mathbf{e}_k$.

In the second step, since the non-zero elements of $\mathbf{B}$ only locate in the kth and $(k+1)$th rows and $\mathbf{q} \in \text{span}[\mathbf{e}_k, \mathbf{e}_{k+1}]$, it indicates $\mathbf{q}$ is the direction with the maximal variance on the span of $\mathbf{e}_k$ and $\mathbf{e}_{k+1}$. Assuming $\mathbf{q} = [0, \cdots, \cos(\gamma), \sin(\gamma), \cdots, 0]^\top$ with $\cos(\gamma)$ and $\sin(\gamma)$ being in the kth and $(k+1)$th coordinates respectively and according to the definition of

principal components, we can find γ by solving the optimization problem

$$\underset{\gamma}{\text{argmax}}: \quad \mathbf{q}^\top \mathbf{Y}\mathbf{Y}^\top \mathbf{q}. \tag{F.1}$$

Plug $\mathbf{q} = [0, \cdots, \cos(\gamma), \sin(\gamma), \cdots, 0]^\top$ into the objective function, and we have

$$\mathbf{q}^\top \mathbf{Y}\mathbf{Y}^\top \mathbf{q} = \begin{bmatrix} \cos(\gamma) \\ \sin(\gamma) \end{bmatrix}^\top \begin{bmatrix} b_{x1} & \frac{1}{2}b_y \\ \frac{1}{2}b_y & b_{x2} \end{bmatrix} \begin{bmatrix} \cos(\gamma) \\ \sin(\gamma) \end{bmatrix}, \tag{F.2}$$

where $b_{x1} = \|\mathbf{b}_k + \mathbf{e}_k\sigma_k\|^2$, $b_{x2} = \|\mathbf{b}_{k+1} + \mathbf{e}_{k+1}\sigma_{k+1}\|^2$, $b_y = 2(\mathbf{b}_k + \mathbf{e}_k\sigma_k)^\top(\mathbf{b}_{k+1} + \mathbf{e}_{k+1}\sigma_{k+1})$, with $\mathbf{b}_k$ and $\mathbf{b}_{k+1}$ being the transpose of the kth and $(k+1)$th rows of $\mathbf{B}$ respectively and $\mathbf{e}_k \in \mathbb{R}^n$, $\mathbf{e}_{k+1} \in \mathbb{R}^n$ being the standard bases.

We can solve (F.2) by computing the first principal component of the middle matrix of the right hand of (F.2). Using the result from equation (4.23), we have $\gamma = 0.5\text{atan2}(b_y, b_x)$, where $b_x = b_{x1} - b_{x2}$. Since the subspace distance is the distance between $\mathbf{q}$ and $\mathbf{e}_k$, it is apparent that the subspace distance is $|\gamma|$. To maximize $|\gamma|$, we first determine the sign of b_y or b_x. We have

$$\begin{aligned}
&\frac{b_x}{\|\mathbf{b}_k + \mathbf{e}_k\sigma_k\| + \|\mathbf{b}_{k+1} + \mathbf{e}_{k+1}\sigma_{k+1}\|} \\
&= \|\mathbf{b}_k + \mathbf{e}_k\sigma_k\| - \|\mathbf{b}_{k+1} + \mathbf{e}_{k+1}\sigma_{k+1}\| \\
&\geq \sigma_k - \|\mathbf{b}_k\| - \sigma_{k+1} - \|\mathbf{b}_{k+1}\| \\
&\geq \sigma_k - \sigma_{k+1} - \sqrt{2}\eta \qquad \text{(F.3)} \\
&> 0, \qquad \text{(F.4)}
\end{aligned}$$

where inequality (F.3) is the result of the energy constraint that $\eta \geq \|\mathbf{B}\|_\text{F} = \sqrt{\|\mathbf{b}_k\|^2 + \|\mathbf{b}_{k+1}\|^2} \geq \frac{1}{\sqrt{2}}(\|\mathbf{b}_k\| + \|\mathbf{b}_{k+1}\|)$, and inequality (F.4) is due to the assumption that $\eta < \frac{\sigma_k - \sigma_{k+1}}{\sqrt{2}}$. In summary, b_x is positive. Using the property of atan2 function, when $b_x > 0$, maximizing $|\gamma|$ is

equivalent to maximizing $|b_y/b_x|$. Thus, we can formulate our problem as

$$\begin{aligned} \max_{\mathbf{b}_k, \mathbf{b}_{k+1}} : \quad & |b_y/b_x| \\ \text{s.t.} \quad & \|[\mathbf{b}_k, \mathbf{b}_{k+1}]\|_{\mathrm{F}} \leq \eta. \end{aligned} \tag{F.5}$$

In the objective function,

$$\begin{aligned} b_y &= 2\big(\mathbf{b}_1^\top \mathbf{b}_2 + (b_{k,k} + \sigma_k) b_{k+1,k} + b_{k,k+1}(b_{k+1,k+1} + \sigma_{k+1})\big), \\ b_x &= \|\mathbf{b}_1\|^2 - \|\mathbf{b}_2\|^2 + (b_{k,k} + \sigma_k)^2 + b_{k,k+1}^2 - b_{k+1,k}^2 - (b_{k+1,k+1} + \sigma_{k+1})^2, \end{aligned}$$

where $\mathbf{b}_1 = [b_{k,1},\, b_{k,2},\, \cdots,\, b_{k,k-1},\, b_{k,k+2},\, \cdots,\, b_{k,n}]^\top$ and $\mathbf{b}_2 = [b_{k+1,1},\, b_{k+1,2},\, \cdots,\, b_{k+1,k-1},\, b_{k+1,k+2},\, \cdots,\, b_{k+1,n}]^\top$ which are the vectors obtained by deleting the kth and $(k+1)$th elements of $\mathbf{b}_k$ and $\mathbf{b}_{k+1}$ respectively. We can change the sign of b_y/b_x by changing the signs of $\mathbf{b}_1, b_{k+1,k}$, and $b_{k,k+1}$. Since both of the values b_y/b_x and $-b_y/b_x$ are obtainable, we can remove the absolute value operation. Thus, our objective can be further simplified to maximize b_y/b_x. To complete the proof of Theorem 4.4, we should further demonstrate that when the optimality of our objective function is obtained, $\mathbf{b}_1$ and $\mathbf{b}_2$ should be vectors with all their entries being zero. To prove that, we examine the objective function further

$$b_y \leq 2\big(\|\mathbf{b}_1\|\|\mathbf{b}_2\| + (b_{k,k} + \sigma_k) b_{k+1,k} + b_{k,k+1}(b_{k+1,k+1} + \sigma_{k+1})\big) \tag{F.6}$$

$$\leq 2\big((b_{k,k} + \sigma_k) b_{k+1,k} + \sqrt{b_{k,k+1}^2 + \|\mathbf{b}_1\|^2}\,(\sqrt{b_{k+1,k+1}^2 + \|\mathbf{b}_2\|^2} + \sigma_{k+1})\big), \tag{F.7}$$

$$b_x \geq (b_{k,k} + \sigma_k)^2 + b_{k,k+1}^2 + \|\mathbf{b}_1\|^2 - b_{k+1,k}^2 - (\sqrt{b_{k+1,k+1}^2 + \|\mathbf{b}_2\|^2} + \sigma_{k+1})^2. \tag{F.8}$$

Inequality (F.6) implies that the optimal value is determined by the norms of $\mathbf{b}_1$ and $\mathbf{b}_2$

instead of their specific values. Inequality (F.7) is true as

$$\begin{aligned}
&\sqrt{b_{k,k+1}^2 + \|\mathbf{b}_1\|^2}(\sqrt{b_{k+1,k+1}^2 + \|\mathbf{b}_2\|^2} + \sigma_{k+1}) \\
&= \sqrt{b_{k,k+1}^2 + \|\mathbf{b}_1\|^2}\sqrt{b_{k+1,k+1}^2 + \|\mathbf{b}_2\|^2} + \sigma_{k+1}\sqrt{b_{k,k+1}^2 + \|\mathbf{b}_1\|^2} \\
&\geq b_{k,k+1}b_{k+1,k+1} + \|\mathbf{b}_1\|\|\mathbf{b}_2\| + \sigma_{k+1}b_{k,k+1} \\
&= \|\mathbf{b}_1\|\|\mathbf{b}_2\| + b_{k,k+1}(b_{k+1,k+1} + \sigma_{k+1}).
\end{aligned}$$

Inequality (F.8) is due to $-(\sqrt{b_{k+1,k+1}^2 + \|\mathbf{b}_2\|^2} + \sigma_{k+1})^2 \leq -\|\mathbf{b_2}\|^2 - (b_{k+1,k+1} + \sigma_{k+1})^2$. The equalities in (F.7) and (F.8) hold when $\|\mathbf{b}_1\| = 0$ and $\|\mathbf{b}_2\| = 0$. This means that, for any feasible solution $(\mathbf{b}_1, \mathbf{b}_2, b_{k,k}, b_{k,k+1}, b_{k+1,k}, b_{k+1,k+1})$ in (F.5), there is another corresponding feasible solution $(\mathbf{0}, \mathbf{0}, b_{k,k}, \sqrt{b_{k,k+1}^2 + \|\mathbf{b}_1\|^2}, b_{k+1,k}, \sqrt{b_{k+1,k+1}^2 + \|\mathbf{b}_2\|^2})$, which has a larger objective value. In conclusion, $\mathbf{b}_1$ and $\mathbf{b}_2$ should be zero vectors when the optimality of (F.5) is obtained. This completes our proof.

Appendix G

Connection Between Asimov Distance and PCR Problem

We first illustrate a connection between the Asimov distance and the projection 2-norm. We then use this connection to establish a connection between the Asimov distance and the PCR problem.

To see the relationship between the Asimov distance and the projection 2-norm, assume $\mathbb{X}$ be the k-dimensional subspace learned from the original data matrix and $\hat{\mathbb{X}}$ be the k-dimensional subspace learned from the modified data matrix. Furthermore, let $\mathbf{P} \in \mathbb{R}^{n \times n}$ be the orthogonal projection onto $\mathbb{X}$ and $\hat{\mathbf{P}} \in \mathbb{R}^{n \times n}$ be the orthogonal projection onto $\hat{\mathbb{X}}$. Then, the Asimov distance between $\mathbb{X}$ and $\hat{\mathbb{X}}$, denoted as $\theta(\mathbb{X}, \hat{\mathbb{X}})$, can also be computed as:

$$\sin\theta = \|\mathbf{P} - \hat{\mathbf{P}}\|_2,$$

where $\|\cdot\|_2$ is the induced 2-norm. Detailed proof can be found in Chapter 2.5 of [67].

Using results in this chapter and the aforementioned relationship between Asimov distance and projection 2-norm, we can perform further analysis on the PCR problem. In particular, let $r_1 = \|\mathbf{y} - \mathbf{y}_1\|$ denote the residual after PCR, where $\mathbf{y}$ is the response vector and $\mathbf{y}_1 = \mathbf{P}\mathbf{y}$ is the projection of $\mathbf{y}$ onto the selected k-dimensional subspace according to

the original feature matrix. Denote $r_2 = \|\mathbf{y} - \mathbf{y}_2\|$ as the residual of PCR after we modify the feature matrix, where $\mathbf{y}_2 = \hat{\mathbf{P}}\mathbf{y}$ is the projection of $\mathbf{y}$ onto the selected k-dimensional subspace after we modify the feature matrix. The following inequality shows that the difference of the two residuals can be bounded by the product of the norm of $\mathbf{y}$ and the projection 2-norm:

$$\begin{aligned}
|r_1 - r_2| &= \big|\|\mathbf{y} - \mathbf{y}_1\| - \|\mathbf{y} - \mathbf{y}_2\|\big| \\
&= \big|\|\mathbf{y} - \mathbf{P}\mathbf{y}\| - \|\mathbf{y} - \hat{\mathbf{P}}\mathbf{y}\|\big| \\
&\leq \|(\mathbf{P} - \hat{\mathbf{P}})\mathbf{y}\| \\
&\leq \|\mathbf{P} - \hat{\mathbf{P}}\|_2 \|\mathbf{y}\| \\
&= \|\mathbf{y}\| \sin\theta.
\end{aligned}$$

As our analysis shows, θ depends on the energy budget and the singular values of the original feature matrix. Hence, given the energy budget and the original data points, we can establish the largest possible change of the residual compared with the original residual.

References

[1] S. G. Finlayson, J. D. Bowers, J. Ito, J. L. Zittrain, A. L. Beam, and I. S. Kohane, "Adversarial attacks on medical machine learning," *Science*, vol. 363, no. 6433, pp. 1287–1289, Mar. 2019.

[2] A. E. Sallab, M. Abdou, E. Perot, and S. Yogamani, "Deep reinforcement learning framework for autonomous driving," *Electronic Imaging*, vol. 2017, no. 19, pp. 70–76, Jan. 2017.

[3] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. International Conference on Learning Representations*, San Diego, CA, May 2015.

[4] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," in *Proc. International Conference on Learning Representations*, Toulon, France, Apr. 2017.

[5] I. Goodfellow, P. McDaniel, and N. Papernot, "Making machine learning robust against adversarial inputs," *Communications of the ACM*, vol. 61, no. 7, pp. 56–66, Jun. 2018.

[6] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," in *Proc. International Conference on Machine Learning*, Edinburgh, Scotland, Jun. 2012, pp. 1807–1814.

[7] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "Badnets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47 230–47 244, Apr. 2019.

[8] S. Alfeld, X. Zhu, and P. Barford, "Data poisoning attacks against autoregressive models," in *Proc. AAAI Conference on Artificial Intelligence*, Phoenix, Arizona, Feb. 2016, pp. 1452–1458.

[9] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li, "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," in *Proc. IEEE Symposium on Security and Privacy*, San Francisco, CA, May 2018, pp. 19–35.

[10] H. Xiao, B. Biggio, G. Brown, G. Fumera, C. Eckert, and F. Roli, "Is feature selection secure against training data poisoning?" in *Proc. International Conference on Machine Learning*, Lille, France, Jul. 2015, pp. 1689–1698.

[11] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein, "Poison frogs! targeted clean-label poisoning attacks on neural networks," in *Proc. Conference on Neural Information Processing Systems*, Montréal, Canada, Dec. 2018, pp. 6106–6116.

[12] L. Lyu, X. He, F. Wu, and L. Sun, "Killing two birds with one stone: Stealing model and inferring attribute from bert-based apis," *arXiv:2105.10909*, May 2021.

[13] X. He, L. Lyu, L. Sun, and Q. Xu, "Model extraction and adversarial transferability, your bert is vulnerable!" in *Proc. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online, Jun. 2021, pp. 2006–2012.

[14] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," *arXiv preprint arXiv:1712.05526*, Dec. 2017.

[15] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. ACM on Asia Conference on Computer and Communications Security*, Abu Dhabi, United Arab Emirates, Apr. 2017, pp. 506–519.

[16] D. Meng and H. Chen, "Magnet: a two-pronged defense against adversarial examples," in *Proc. ACM SIGSAC Conference on Computer and Communications Security*, Dallas, TX, Oct. 2017, pp. 135–147.

[17] X. Yan and X. Su, *Linear regression analysis: theory and computing*. World Scientific, 2009.

[18] G. Papageorgiou, P. Bouboulis, and S. Theodoridis, "Robust linear regression analysis— a greedy approach," *IEEE Transactions on Signal Processing*, vol. 63, no. 15, pp. 3872–3887, Aug. 2015.

[19] X. Jiang, W. Zeng, H. C. So, A. M. Zoubir, and T. Kirubarajan, "Beamforming via nonconvex linear regression," *IEEE Transactions on Signal Processing*, vol. 64, no. 7, pp. 1714–1728, Apr. 2016.

[20] J. Chien and J. Chen, "Recursive Bayesian linear regression for adaptive classification," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 565–575, Feb. 2009.

[21] T. Gustafsson and B. D. Rao, "Statistical analysis of subspace-based estimation of reduced-rank linear regressions," *IEEE Transactions on Signal Processing*, vol. 50, no. 1, pp. 151–159, Jan. 2002.

[22] J. H. McDonald, *Handbook of biological statistics*. Sparky House Publishing, 2009.

[23] O. E. Barndorff-Nielsen and N. Shephard, "Econometric analysis of realized covariation: High frequency based covariance, regression, and correlation in financial economics," *Econometrica*, vol. 72, no. 3, pp. 885–925, May 2004.

[24] C. J. ter Braak and S. Juggins, "Weighted averaging partial least squares regression (WA-PLS): an improved method for reconstructing environmental variables from species assemblages," in *Proc. International Diatom Symposium*, Renesse, The Netherlands, Aug. 1993, pp. 485–502.

[25] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[26] F. Li, L. Lai, and S. Cui, "On the adversarial robustness of subspace learning," *IEEE Transactions on Signal Processing*, vol. 68, pp. 1470–1483, Mar. 2020.

[27] D. L. Pimentel-Alarcón, A. Biswas, and C. R. Solís-Lemus, "Adversarial principal component analysis," in *Proc. IEEE International Symposium on Information Theory*, Aachen, Germany, Jun. 2017, pp. 2363–2367.

[28] B. Biggio, B. Nelson, and P. Laskov, "Support vector machines under adversarial label noise," in *Proc. Asian Conference on Machine Learning*, Taoyuan, Taiwan, Nov. 2011, pp. 97–112.

[29] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: A simple and accurate method to fool deep neural networks," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, Jun. 2016, pp. 2574–2582.

[30] B. Biggio, A. Demontis, A. Paudice, V. Wongrassamee, E. C. Lupu, and F. Roli, "Towards poisoning of deep learning algorithms with back-gradient optimization," in *Proc. ACM Workshop on Artificial Intelligence and Security*, Dallas, TX, Oct. 2017, pp. 27–38.

[31] H. Kwon, Y. Kim, H. Yoon, and D. Choi, "Selective audio adversarial example in evasion attack on speech recognition system," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 526–538, Jun. 2019.

[32] D. Li and Q. Li, "Adversarial deep ensemble: Evasion attacks and defenses for malware detection," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3886–3900, Jun. 2020.

[33] B. Flowers, R. M. Buehrer, and W. C. Headley, "Evaluating adversarial evasion attacks in the context of wireless communications," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1102–1113, Aug. 2019.

[34] S. Mei and X. Zhu, "Using machine teaching to identify optimal training-set attacks on machine learners," in *Proc. AAAI Conference on Artificial Intelligence*, Austin, Texas, Jan. 2015, pp. 2871–2877.

[35] F. Li, L. Lai, and S. Cui, "On the adversarial robustness of linear regression," in *Proc. IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, Espoo, Finland, Sep. 2020, pp. 1–6.

[36] ——, "Optimal feature manipulation attacks against linear regression," *IEEE Transactions on Signal Processing*, 2021, under review.

[37] M. Dash and H. Liu, "Feature selection for classification," *Intelligent Data Analysis*, vol. 1, no. 3, pp. 131–156, Jan. 1997.

[38] F. D. Mandanas and C. L. Kotropoulos, "Subspace learning and feature selection via orthogonal mapping," *IEEE Transactions on Signal Processing*, vol. 68, pp. 1034–1047, Jan. 2020.

[39] C. Furlanello, S. Merler, and G. Jurman, "Combining feature selection and DTW for time-varying functional genomics," *IEEE Transactions on Signal Processing*, vol. 54, no. 6, pp. 2436–2443, Jun. 2006.

[40] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[41] M. Tan, I. W. Tsang, and L. Wang, "Matching pursuit LASSO part I: Sparse recovery over big dictionary," *IEEE Transactions on Signal Processing*, vol. 63, no. 3, pp. 727–741, Feb. 2015.

[42] L. M. Butcher and S. Beck, "Probe lasso: a novel method to rope in differentially methylated regions with 450k DNA methylation data," *Methods*, vol. 72, pp. 21–28, Jan. 2015.

[43] Y. Zhang, F. Ma, and Y. Wang, "Forecasting crude oil prices with a large set of predictors: Can LASSO select powerful predictors?" *Journal of Empirical Finance*, vol. 54, pp. 97–117, Dec. 2019.

[44] D. Yang and W. Bao, "Group lasso-based band selection for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 12, pp. 2438–2442, Nov. 2017.

[45] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, Feb. 2006.

[46] X. Lv, G. Bi, and C. Wan, "The group lasso for stable recovery of block-sparse signal representations," *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1371–1382, Jan. 2011.

[47] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, "A sparse-group LASSO," *Journal of Computational and Graphical Statistics*, vol. 22, no. 2, pp. 231–245, May 2013.

[48] B. Zhang, J. Geng, and L. Lai, "Multiple change-points estimation in linear regression models via sparse group LASSO," *IEEE Transactions on Signal Processing*, vol. 63, no. 9, pp. 2209–2224, May 2015.

[49] J. Jeong and C. Kim, "Effect of outliers on the variable selection by the regularized regression," *Communications for Statistical Applications and Methods*, vol. 25, no. 2, pp. 235–243, Mar. 2018.

[50] P.-L. Loh and M. J. Wainwright, "High-dimensional regression with noisy and missing data: Provable guarantees with non-convexity," in *Advances in Neural Information Processing Systems*, Granada, Spain, Dec. 2011, pp. 2726–2734.

[51] F. Li, L. Lai, and S. Cui, "On the adversarial robustness of feature selection using LASSO," in *Proc. IEEE International Workshop on Machine Learning for Signal Processing*, Espoo, Finland, Sep. 2020, pp. 1–6.

[52] ——, "On the adversarial robustness of LASSO based feature selection," *IEEE Transactions on Signal Processing*, Jun. 2021, under review.

[53] Y. Li, W. Dai, J. Zou, H. Xiong, and Y. F. Zheng, "Structured sparse representation with union of data-driven linear and multilinear subspaces model for compressive video sampling," *IEEE Transactions on Signal Processing*, vol. 65, no. 19, pp. 5062–5077, Oct. 2017.

[54] J. Xin, N. Zheng, and A. Sano, "Subspace-based adaptive method for estimating direction-of-arrival with luenberger observer," *IEEE Transactions on Signal Processing*, vol. 59, no. 1, pp. 145–159, Jan. 2011.

[55] Y. Shen, M. Mardani, and G. B. Giannakis, "Online categorical subspace learning for sketching big data with misses," *IEEE Transactions on Signal Processing*, vol. 65, no. 15, pp. 4004–4018, Aug. 2017.

[56] H. Guo, C. Qiu, and N. Vaswani, "An online algorithm for separating sparse and low-dimensional signal sequences from their sum," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4284–4297, Aug. 2014.

[57] R. Otazo, E. J. Candès, and D. K. Sodickson, "Low-rank plus sparse matrix decomposition for accelerated dynamic MRI with separation of background and dynamic components," *Magnetic Resonance in Medicine*, vol. 73, no. 3, pp. 1125–1136, Apr. 2015.

[58] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[59] M. Mardani, G. Mateos, and G. B. Giannakis, "Dynamic anomalography: Tracking network anomalies via sparsity and low rank," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 1, pp. 50–66, Feb. 2012.

[60] H. Guo and N. Vaswani, "Video denoising via online sparse and low-rank matrix decomposition," in *Proc. IEEE Statistical Signal Processing Workshop*, Palma de Mallorca, Spain, Jun. 2016, pp. 1–5.

[61] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM*, vol. 58, no. 3, pp. 11:1–11:37, Jun. 2011.

[62] D. Hsu, S. M. Kakade, and T. Zhang, "Robust matrix decomposition with sparse corruptions," *IEEE Transactions on Information Theory*, vol. 57, no. 11, pp. 7221–7234, Jun. 2011.

[63] C. Qiu, N. Vaswani, B. Lois, and L. Hogben, "Recursive robust PCA or recursive sparse recovery in large but structured noise," *IEEE Transactions on Information Theory*, vol. 60, no. 8, pp. 5007–5039, Jun. 2014.

[64] Y. Chen, H. Xu, C. Caramanis, and S. Sanghavi, "Robust matrix completion and corrupted columns," in *Proc. International Conference on Machine Learning*, Bellevue, Washington, Jun. 2011, pp. 873–880.

[65] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, Jun. 2018, pp. 1625–1634.

[66] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, and W. Zhou, "Hidden voice commands," in *Proc. USENIX Security Symposium*, Austin, TX, Aug. 2016, pp. 513–530.

[67] G. H. Golub and C. F. Van Loan, *Matrix computations*. The Johns Hopkins University Press, 2013.

[68] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM journal on Matrix Analysis and Applications*, vol. 20, no. 2, pp. 303–353, Apr. 1998.

[69] A. Weinstein, "Almost invariant submanifolds for compact group actions," *Journal of the European Mathematical Society*, vol. 2, no. 1, pp. 53–86, Mar. 2000.

[70] T. T. Georgiou and M. C. Smith, "Optimal robustness in the gap metric," *IEEE Transactions on Automatic Control*, vol. 35, no. 6, pp. 673–686, Jun. 1990.

[71] G. Vinnicombe, "Frequency domain uncertainty and the graph topology," *IEEE Transactions on Automatic Control*, vol. 38, no. 9, pp. 1371–1383, Sep. 1993.

[72] L. Qui and E. Davison, "Feedback stability under simultaneous gap metric uncertainties in plant and controller," *Systems & Control Letters*, vol. 18, no. 1, pp. 9–22, Jan. 1992.

[73] C. He and J. M. Moura, "Robust detection with the gap metric," *IEEE Transactions on Signal Processing*, vol. 45, no. 6, pp. 1591–1604, Jun. 1997.

[74] P. A. Absil, A. Edelman, and P. Koev, "On the largest principal angle between random subspaces," *Linear Algebra and its applications*, vol. 414, no. 1, pp. 288–294, Apr. 2006.

[75] L. Lai and E. Bayraktar, "On the adversarial robustness of robust estimators," *IEEE Transactions on Information Theory*, vol. 66, no. 8, pp. 5097–5109, Aug. 2020.

[76] F. Li, L. Lai, and S. Cui, "On the adversarial robustness of subspace learning," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Brighton, UK, May 2019, pp. 2477–2481.

[77] Y. Li, F. Li, L. Lai, and J. Wu, "On the adversarial robustness of principal component analysis," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Toronto, Canada, Jun. 2021, pp. 3695–3699.

[78] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge University Press, 2012.

[79] A. Beck and M. Teboulle, "On minimizing quadratically constrained ratio of two quadratic functions," *Journal of Convex Analysis*, vol. 17, no. 3, pp. 789–804, 2010.

[80] R. J. Stern and H. Wolkowicz, "Indefinite trust region subproblems and nonsymmetric eigenvalue perturbations," *SIAM Journal on Optimization*, vol. 5, no. 2, pp. 286–313, May 1995.

[81] A. Ben-Tal and M. Teboulle, "Hidden convexity in some nonconvex quadratically constrained quadratic programming," *Mathematical Programming*, vol. 72, no. 1, pp. 51–63, Jan. 1996.

[82] Y. Ye and S. Zhang, "New results on quadratic minimization," *SIAM Journal on Optimization*, vol. 14, no. 1, pp. 245–267, 2003.

[83] J. B. Lasserre, "Global optimization with polynomials and the problem of moments," *SIAM Journal on Optimization*, vol. 11, no. 3, pp. 796–817, 2001.

[84] M. Laurent, "Sums of squares, moment matrices and optimization over polynomials," in *Emerging Applications of Algebraic Geometry*. Springer, 2009, pp. 157–270.

[85] T. Weisser, J. B. Lasserre, and K.-C. Toh, "Sparse-BSOS: a bounded degree SOS hierarchy for large scale polynomial optimization with sparsity," *Mathematical Programming Computation*, vol. 10, no. 1, pp. 1–32, 2018.

[86] M. J. Wainwright and M. I. Jordan, "Log-determinant relaxation for approximate inference in discrete markov random fields," *IEEE transactions on signal processing*, vol. 54, no. 6, pp. 2099–2109, Jun. 2006.

[87] L. Porkolab and L. Khachiyan, "On the complexity of semidefinite programs," *Journal of Global Optimization*, vol. 10, no. 4, pp. 351–365, 1997.

[88] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," *Technical University of Denmark*, 2008.

[89] A. Beck and M. Teboulle, "A convex optimization approach for minimizing the ratio of indefinite quadratic functions over an ellipsoid," *Mathematical Programming*, vol. 118, no. 1, pp. 13–35, Apr. 2009.

[90] F. Rendl, "A matlab toolbox for semidefinite programming," *The program can be found at ftp://orion. uwaterloo. ca/pub/henry/teaching/co769g*, 1994.

[91] J. Lofberg, "Yalmip: A toolbox for modeling and optimization in matlab," in *Proc. International Conference on Robotics and Automation*, New Orleans, LA, Apr. 2004, pp. 284–289.

[92] A. Beck, A. Ben-Tal, and M. Teboulle, "Finding a global optimal solution for a quadratically constrained fractional quadratic problem with applications to the regularized total least squares," *SIAM Journal on Matrix Analysis and Applications*, vol. 28, no. 2, pp. 425–445, 2006.

[93] R. G. Bartle and D. R. Sherbert, *Introduction to real analysis*. Wiley New York, 2000.

[94] L. Grippo and M. Sciandrone, "On the convergence of the block nonlinear Gauss–Seidel method under convex constraints," *Operations Research Letters*, vol. 26, no. 3, pp. 127–136, Apr. 2000.

[95] O. Akbilgic, H. Bozdogan, and M. E. Balaban, "A novel hybrid RBF neural networks model as a forecaster," *Statistics and Computing*, vol. 24, no. 3, pp. 365–375, May 2014.

[96] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[97] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, Jan. 2014.

[98] A. L. Dontchev and R. T. Rockafellar, "Implicit functions and solution mappings," *Springer Monographs in Mathematics. Springer*, vol. 208, Feb. 2009.

[99] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale ℓ_1-regularized least squares," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 606–617, Dec. 2007.

[100] T.-T. Lu and S.-H. Shiou, "Inverses of $2\times$ 2 block matrices," *Computers & Mathematics with Applications*, vol. 43, no. 1-2, pp. 119–129, 2002.

[101] L. Condat, "Fast projection onto the simplex and the ℓ_1 ball," *Mathematical Programming*, vol. 158, no. 1-2, pp. 575–585, Sep. 2015.

[102] S. Liu, Y. D. Zhang, T. Shan, S. Qin, and M. G. Amin, "Structure-aware bayesian compressive sensing for frequency-hopping spectrum estimation," in *Compressive Sensing V: From Diverse Modalities to Big Data Analytics*, vol. 9857. International Society for Optics and Photonics, 2016, p. 98570N.

[103] S. Liu, Y. D. Zhang, T. Shan, and R. Tao, "Structure-aware bayesian compressive sensing for frequency-hopping spectrum estimation with missing observations," *IEEE Transactions on Signal Processing*, vol. 66, no. 8, pp. 2153–2166, 2018.

[104] J. Fang, Y. Shen, H. Li, and P. Wang, "Pattern-coupled sparse bayesian learning for recovery of block-sparse signals," *IEEE Transactions on Signal Processing*, vol. 63, no. 2, pp. 360–372, 2014.

[105] P.-Y. Chen and I. W. Selesnick, "Group-sparse signal denoising: non-convex regularization, convex optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 13, pp. 3464–3478, Jul. 2014.

[106] Q. Zhao, W. X. Li, X. Jiang, J. Lv, J. Lu, and T. Liu, "Functional brain networks reconstruction using group sparsity-regularized learning," *Brain Imaging and Behavior*, vol. 12, no. 3, pp. 758–770, Jun. 2018.

[107] J. Ziniel and P. Schniter, "Dynamic compressive sensing of time-varying signals via approximate message passing," *IEEE Transactions on Signal Processing*, vol. 61, no. 21, pp. 5270–5284, Nov. 2013.

[108] V. Roth and B. Fischer, "The group-lasso for generalized linear models: uniqueness of solutions and efficient algorithms," in *Proc. International Conference on Machine Learning*, Helsinki, Finland, Jul. 2008, pp. 848–855.

[109] S. Chatterjee, K. Steinhaeuser, A. Banerjee, S. Chatterjee, and A. Ganguly, "Sparse group LASSO: Consistency and climate applications," in *Proc. SIAM International Conference on Data Mining*, Anaheim, CA, Apr. 2012, pp. 47–58.

[110] L. Zhao, Q. Hu, and W. Wang, "Heterogeneous feature selection with multi-modal deep neural networks and sparse group LASSO," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 1936–1948, Nov. 2015.

[111] J. H. Kalivas, "Two data sets of near infrared spectra," *Chemometrics and Intelligent Laboratory Systems*, vol. 37, no. 2, pp. 255–259, Jun. 1997.

[112] P. Stoica and K. C. Sharman, "Maximum likelihood methods for direction-of-arrival estimation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 7, pp. 1132–1143, Jul. 1990.

[113] T.-J. Shan, M. Wax, and T. Kailath, "On spatial smoothing for direction-of-arrival estimation of coherent signals," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 4, pp. 806–811, Aug. 1985.

[114] E. Kalnay, M. Kanamitsu, R. Kistler, W. Collins, D. Deaven, L. Gandin, M. Iredell, S. Saha, G. White, J. Woollen *et al.*, "The NCEP/NCAR 40-year reanalysis project," *Bulletin of the American Meteorological Society*, vol. 77, no. 3, pp. 437–472, 1996.

[115] R. Zimmermann, "A geometric approach to subspace updates and orthogonal matrix decompositions under rank-one modifications," *Mathematics of Computation*, vol. 90, no. 328, pp. 671–688, Oct. 2020.

[116] R. C. Thompson, "The behavior of eigenvalues and singular values under perturbations of restricted rank," *Linear Algebra and its Applications*, vol. 13, no. 1-2, pp. 69–78, 1976.

[117] J. E. Jackson, *A user's guide to principal components*. John Wiley & Sons, 2005, vol. 587.

[118] D. Henrion and J. B. Lasserre, "Detecting global optimality and extracting solutions in gloptipoly," in *Positive Polynomials in Control*. Springer, Apr. 2005, pp. 293–310.