New perspectives on optimization: combating data poisoning, solving Euclidean optimization and learning minimax optimal estimators

Kartik Gupta

August 2024
CMU-ML-24-107

Machine Learning Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:

Pradeep Ravikumar (chair)	Carnegie Mellon University
Virginia Smith	Carnegie Mellon University
Stephen D Miller	Rutgers University
Ramarathnam Venkatesan	Microsoft

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

This research was sponsored by: National Science Foundation awards 1664720 and 1934584, Office of Naval Research award N000141812861, and a contract from Accenture.

Keywords: optimization, stochastic optimization, random walks, non-euclidean optimization, robustness, data poisoning, grassmannians, multinomial manifold, neural networks, deep learning, black box, minimax estimators, online learning

To my family,
for letting me choose my own journey.

Abstract

Robustness properties of optimization techniques are a fundamental requirement for modern ML applications. In an environment where vast quantities of data is scraped from the Internet or collected from varied sources, curating clean data has become an increasingly intractable problem. Sophisticated adversaries can easily bypass simple data cleaning methods to poison training data in order to influence the training procedure. Many attacks exist which can construct a backdoor in the learned model enabling maliciously targeted predictions during deployment. Some of the advanced attacks even claim broad generality, wherein, corruptions produced for one model can be successfully used to deploy an attack for another. This is a problem which models of all complexities have to contend with in the real world. To address this issue, existing research has focused on providing specialized solutions for specific model classes. This puts an enormous burden on the practitioner to keep pace with the rapid progress of research. Moreover, a lot of techniques treat the two problems of optimization and handling poisoned data separately.

In the first part of this thesis, we take a unified approach in which we provide a single optimization technique that can be used to train any ML model and which also shows impressive fortitude against data poisoning attacks. We experimentally evaluate our study on a wide class of ML models and provide a theoretical analysis for convergence as well as a mathematical understanding of what enables robustness against data poisoning for our techniques. For a given optimization problem, our technique constructs a sequence of smaller-dimensional optimization problems. It assumes access to a black box that can solve these smaller-dimensional problems and the solution of the last problem in the sequence is the desired model.

We continue with the theme of developing algorithms that use cleverly constructed black-box solvers in the second and third parts of this thesis. In the second part, we pose the problem of solving an Euclidean optimization problem as one that of solving an optimization problem on a manifold (specifically, the Grassmannian and the Multinomial manifold). This transforms an optimization problem from a given dimension to a sequence of problems in a smaller dimension. For a class of optimization problems which are defined using a data matrix, we also develop a technique which reduces the row dimension of the data matrix to construct the sequence of problems to solve. These techniques provide a very novel perspective on optimization problems in the Euclidean space and have the potential to inspire future developments of optimization procedures with robustness and privacy properties.

In the third part of the thesis, we study online optimization algorithms with access to black-box solvers for minimization and maximization procedures. These algorithms can be used to find the Nash Equilibrium of two-player zero-sum games. We formulate the problem of finding minimax optimal estimators

as that of finding a Nash Equilibrium of a two-player zero-sum game. This helps us bypass the mathematical complexity of constructing minimax optimal estimators, which usually involve a lot of problem-specific analysis and development of new theoretical tools, to directly learn them as neural networks by solving the stated two-player game. Using this technique, we are able to construct minimax optimal estimators for a class of fundamental statistical estimation problems for which optimal estimators have not been known prior to this work.

In summary, in this thesis we develop techniques which have broad applicability within their domains (the domains being robust optimization, Euclidean optimization, and learning minimax estimators) by leveraging optimization procedures which have access to appropriate black boxes.

Acknowledgments

I have been very fortunate during my Ph.D. journey for the support, guidance, and companionship that I received at every stage of it. I would like to express my gratitude to everyone who helped me through this arduous but very fulfilling journey.

To begin with, I would like to thank my advisor Pradeep Ravikumar. He shaped my taste in research in the early years of my Ph.D. and helped me make the transition to becoming a researcher. Several people who I got introduced to as part of the projects I worked on in his research group ended up becoming valued friends and mentors later in my years at CMU. He was extremely supportive with my choice of research direction and was very patient over the years when I was digging deep on one problem, without necessarily having something very concrete to show for it. It helped me learn things on my own terms and take charge of my own growth as a researcher; for this, I am very grateful.

I would like to thank Ramarathnam Venkatesan, who gave me a very interesting problem to work on which occupied most of my time as a graduate student. He taught me a lot of wonderful mathematics and always believed in my ability to deliver good work.

I would like to thank Stephen D. Miller for being a great mentor. He always found time to fill the gaps in my mathematical knowledge during our discussions. He was always encouraging in whatever ideas I decided to pursue and enthusiastically cheered me on in all my discoveries. His insistence on precision, his deep knowledge of mathematics, and his ability to quickly share the core insights of any complex field is something I will always remember. He is someone I will always look up to.

I would like to thank Bhargav Narayanan for providing valuable advice and a fair, open and empathetic ear to the many problems I brought to him.

I would like to thank Arun Sai Suggala for his guidance in the early years of my PhD. Especially for teaching me how to actually make optimization algorithms converge in practice.

I would like to thank Dhivya Eswaran, Varun Gangal, Abhinav Garlapati and Raj Sengupta, for being excellent friends, especially in the first year of my Ph.D. They made my move from India to the US seamless.

I would like to thank Leqi Liu, Biswajit Paria, Giulio Zhou, Eyan Noronha, Stefania La Vattiata and Kin Gutierrez without whom I could not have survived either Pittsburgh or the PhD journey. They are the gems I collected during this journey of my life and will hold on to for the longest time to come.

I would also like to thank Adarsh Prasad, Chirag Gupta, Bingbin Liu, Sebastian Caldas, Chih-Kuan Yeh, Helen Zhou, Chen Dan, Himanshu Zade and Jin Li for being very good friends throughout the past seven years. To this list I would also like to add my friends Aishwarya Padmakumar, Aarati Kakaraparthy and Aditi Raghunathan, with whom I have shared an even longer friendship.

I would like to thank Diane Stidle and Dorothy Holland-Minkley for taking such good care of all graduate students in the department. MLD will not be the place it is without them.

Last but not the least, I would like to thank my mother, my father, my sister and my brother-in-law. They have always believed in me and supported me in my journey. They have stood by me in difficult times and have gone out of their way to take care of me. I would especially like to thank my sister, for her wisdom, love and compassion. I was surprised to see how strong bonds made in childhood can carry over in adult life when, after a decade, I got to spend some extended amount of time with her in Pittsburgh. I think spending this time was the final thing I needed to finish my PhD, as it reminded me that so much more is waiting for me on the other side of graduation.

Contents

List of Figures

List of Tables

1 | Introduction

Modern ML systems are built with large quantities of data. These data are collected from a number of sources and can be measured from giga-to-terabytes [75]. The performance of trained models is crucially dependent on the quality of the training data used [31, 42, 90]. Due to the sheer size of the training data, curating it can be extremely challenging. One cannot trust the sources where the data is collected from entirely and widely used methods of data collection like scraping it from publicly available sources leave these datasets vulnerable to attacks from adversaries, especially in the form of injecting small amounts of adversarially chosen data. This can either throw off the learning process to decrease the quality of the model learned [33, 36] or more maliciously inject backdoors which can enable the adversary to manipulate the predictions of the model during deployment [24, 76, 92]. See [25] for a comprehensive survey on data poisoning strategies in the research literature.

Combating such challenges involves deploying preventive techniques at every stage of the ML pipeline. From data collection to data sanitization, model selection, robust training algorithms, and building guardrails around the delpoyed model. While these preventive techniques are necessary to keep lazy adversaries at bay, they usually tend to be heuristics and can be side-stepped by the next clever idea. Especially techniques which address the data in bulk tend to lack the nuance with which a motivated adversary can work, since at these stages it can be difficult to distinguish malicious data from normal data except by human inspection. Interestingly, techniques which can bypass human inspection have also been proposed in the literature. Hence, it is important to develop a mathematical understanding of how and when a small amount of data can negatively affect the training process or the learned model. Using this, techniques can be provided with a sound mathematical understanding of how to build robust models. The model selection and training algorithms provide the most fertile ground for this task. Among these two, we choose to work with the training algorithm, since it has the added benefit of being generally applicable to a wide range of real-world scenarios and mixing well with other techniques that might already be in use. The proposed training algorithm works across the range of ML models with which a practitioner might want to work in a given scenario. This forms the first part of the thesis. In this part:

1. In chapter 2, we present new techniques for the building and analyzing of robust stochastic optimization algorithms. To solve the given d-dimensional optimization problem, our technique generates a sequence of random k-dimensional subproblems, where $k < d$, and solves them instead. Unlike traditional optimization analysis which exploits structural assumptions like convexity, Lipschitzness or Polyak-Lojasiewicz

criterion of the loss function to obtain convergence rates, our analysis only uses the geometrical structure of the randomness used in the algorithm. This offers a wider applicability to our approach than traditional methods, and indeed it works for all smooth loss functions. Moreover, our analysis identifies an important parameter of the loss function, which we call the gap parameter. This parameter dictates the convergence rates of our algorithm. We experimentally study the algorithm on linear regression, logistic regression, SVMs and neural networks. Using these experiments, we argue that the gap parameter also controls the robustness of the solutions obtained by our algorithm in the presence of noise in the training data. A modified algorithm which can control the effect of noise on it's output is presented as well. Finally, in this chapter we discuss how the choice of k affects the convergence and robustness of our algorithm.

As stated above, the main technique of the first part of this thesis works by breaking a given optimization problem into a sequence of smaller-dimensional optimization problems. We develop on this idea further in the second part of the thesis. In this part, we formulate the problem of solving an optimization problem in an Euclidean space as that of solving one on a non-Euclidean manifold. Specifically,

2. In chapter 3, we reformulate the problem of solving a Euclidean optimization problem as one that of solving an optimization problem on a Grassmannian. A Grassmannian is a Riemannian manifold parameterized by two positive integers d and k. Every point in this manifold represents a k-dimensional subspace of the d-dimensional Euclidean space. We transform an Euclidean optimization problem to one on a Grassmannian by defining an objective function on the Grassmannian whose value at every point is the minimum value of the function on the Euclidean space when restricted to the subspace corresponding to that point. We then provide formulae and convergence guarantees for gradient descent of this newly defined objective on the Grassmannian.

3. In chapter 4, we take a similar approach as that of the previous chapter but with a different manifold. In this chapter, we work with Euclidean optimization problems of a specific kind which are defined using a data matrix. For these kind of optimization problems we define a corresponding optimization problem on the multinomial manifold. The multinomial manifold is a Riemannian manifold parameterized by two positive integers n and m. Every point of this manifold is a matrix of size $n \times m$ with positive entries and whose every column sums to 1. We provide gradient descent formulae as well as convergence guarantees for the reformulated objective.

The aim of this section is not necessarily to push the state-of-the-art in terms of providing techniques that beat the existing techniques on a certain benchmark. Instead, the aim is to provide a new mathematical perspective on age-old problems to inspire future research directions.

The main algorithmic theme connecting the last two sections was that of using access to a certain well-designed black-box solver in order to solve a bigger problem at hand. By making this black-box assumption, we open doors for innovating new techniques. This connects the work from the last two sections to the work in the next section. In the last part of the thesis, we provide algorithmic ideas for the design of minimax optimal estimators, a fundamental problem in the field of statistics.

Figure 1.1: Clean label backdoor attack image taken from [100]. The backdoor attack is added to the image on the right lower corner with the attack increasing in intensity starting with no attack from left to right.

4. In chapter 5, we consider the problem of designing minimax estimators for estimating the parameters of a probability distribution. Unlike classical approaches such as the MLE and minimum distance estimators, we consider an algorithmic approach for constructing such estimators. We view the problem of designing minimax estimators as finding a mixed strategy Nash equilibrium of a zero-sum game. By leveraging recent results in online learning with non-convex losses, we provide a general algorithm for finding a mixed-strategy Nash equilibrium of general non-convex non-concave zero-sum games. Our algorithm requires access to two subroutines: (a) one which outputs a Bayes estimator corresponding to a given prior probability distribution, and (b) one which computes the worst-case risk of any given estimator. Given access to these two subroutines, we show that our algorithm outputs both a minimax estimator and a least favorable prior. To demonstrate the power of this approach, we use it to construct provably minimax estimators for classical problems such as estimation in the finite Gaussian sequence model, and linear regression. Despite being a well-studied problem, most of the approaches for constructing minimax estimators are problem specific and usually do not extend or generalize to other problems. Often, the process of designing minimax estimators is considered to be an art, as there is no single technique for coming up with these estimators which works for all problems. In this work, we provide algorithmic approaches for constructing minimax estimators. The key advantage of our approach is that it is not problem specific and can be used to construct minimax estimators for general problems. It requires access to two optimization sub-routines. For problems where these sub-routines can be implemented efficiently, our algorithm provides a computationally efficient technique to construct minimax estimators. This chapter is written by referencing the work [43].

1.1 Background

In this section, we provide some mathematical background for the research problems addressed in the thesis.

1.1.1 Data Poisoning

Data poisoning in machine learning refers to the process of modifying the training data, used to train a machine learning model by an adversary to disrupt its learning process. It

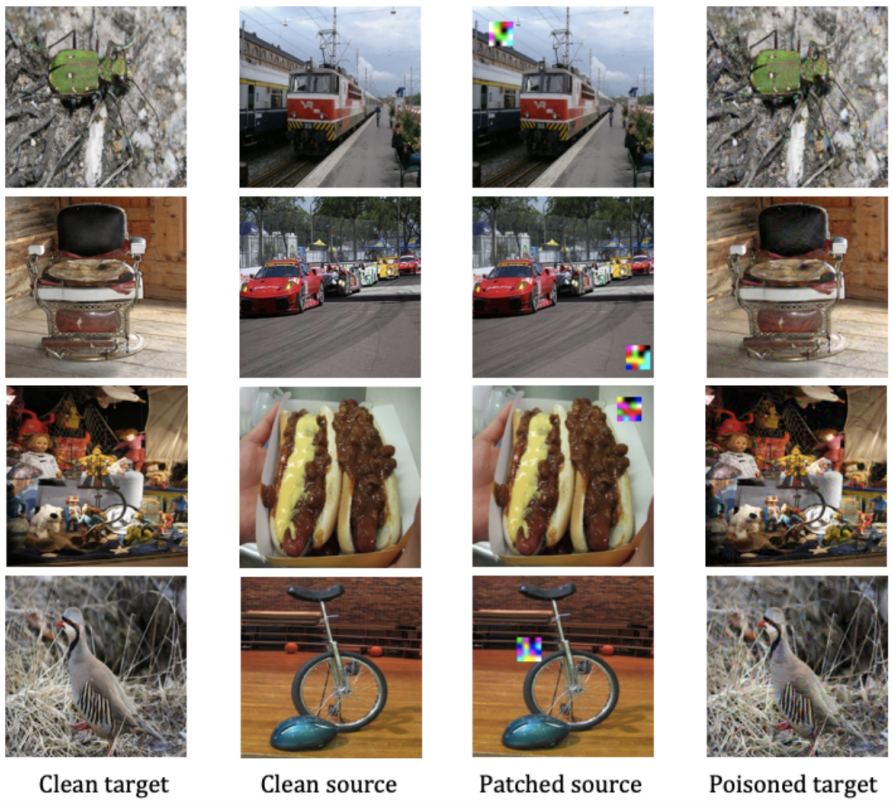


Figure 1.2: Hidden trigger backdoor attack image taken from [100]. The backdoor attack constructs poisoned images which are close to their source image visually but are close to a patched version of an image in a target class in some feature space. The first column in the figure are target images that the adversary wants to poison. The second column are the source images that the adversary wants the corresponding poisoned images to share the class with. The third column is a patched version of the source images and the fourth column are the poisoned targets generated by the attack. The poisoned targets are intended by the attacked to be classified as per the corresponding source image by the attacked classifier.

usually involves either a carefully designed noise added to the existing dataset or a set of cleverly designed new data points. Mathematically, these two ways can be formulated in the same manner as adding noise to the training dataset. Many data poisoning attacks have been proposed in the machine learning literature, and indeed even small amounts of cleverly chosen noise have been shown to produce a large impact on the training of the models. Frameworks have been proposed to classify and study the various attacks and countermeasures that the community has developed, such as [25] and [98]. We give an overview of the attacks against which we tested our techniques.

Attack against Logistic Regression models. We consider the attack presented in [27]. The objective of this attack is to launch a kind of denial of service attack by adding noise to the training data which makes it difficult to learn a meaningful classifier. In this attack, an adversary constructs a bi-level optimization problem to generate the poisoned samples. The bi-level optimization consists of an outer maximization procedure which

finds poisoned samples that maximize the loss desired by the adversary, and an inner minimization procedure which finds the optimal solution over the corrupted data. The inner optimization models the learning that can be performed on the poisoned training data while the outer optimization models how well the adversary does for the given poisoning. This bi-level optimization problem is then solved using projected gradient-ascent to get the poisoned samples. See [27] for more details.

Attack against Support Vector Machines. We consider the attack presented in [13]. The objective and approach of this attack is similar to that presented for Logistic Regression. The attack is intended to increase the training loss and is constructed by solving a bi-level optimization problem using projected gradient ascent.

Attack against Deep Learning models. We consider the attack presented in [100] and [93]. These are backdoor attacks where the adversary places a pattern of pixels cleverly over a subset of the training images with the intention of invoking a desired response from the model when the same pattern of pixels is found on a test image. For example, in a digit recognition setting, an adversary may be interested in classifying any image with the desired pixel pattern present on it as an image of a 0 instead of the actual number in the image. This enables the adversary to manipulate the model to get the desired responses out of the system and hence bypass the model entirely.

In [100], they propose a clean label backdoor attack in which the special pixels are added only to the images that belong to the desired class, for example in the previous scenario the adversary would only add the pixels to images corresponding to a 0 in the training data. This has the added benefit of the attack passing some rudimentary checks like human inspection. We present an example of this attack in Figure 1.1 which is taken from [100]. In [93], they propose a method which constructs poisoned images by solving an optimization problem defined to keep the poisoned image close to it's source visually while moving it close to a patched version of an image of desired target class in the feature space. We present an examples of this attack in Figure 1.2 which is taken from [93].

1.1.2 Euclidean and non-Euclidean optimization

Optimization refers to the task of finding a solution that minimizes or maximizes a given function. In Euclidean optimization, the function is defined over the Euclidean space of a certain dimension d. Let $f : \mathbb{R}^d \to \mathbb{R}$ be the function that we want to minimize, a popular and generally applicable method to do so is gradient descent. In gradient descent, one starts out with an initial guess for the minimizing solution and improves upon it by moving in the direction opposite to the gradient of the function at that point, repeating the same procedure at the new solution obtained, and so on. Let $\mathbf{x}_0 \in \mathbb{R}^d$ be the initial guess for the minimizing solution, then at any step $i \geq 1$ of the gradient descent step we update $\mathbf{x}_0$ using,

$$\mathbf{x}_i = \mathbf{x}_{i-1} - \eta \nabla f(\mathbf{x}_{i-1})$$

where η is the learning rate of the algorithm which decides how much to move in one step of the algorithm. Many theoretical results exist that show convergence of this procedure to global minima, local minima, or saddle points of f depending on it's mathematical properties [11]. Variants of gradient descent are used to train most modern Machine Learning

systems. For our purposes, we will only be dealing with the basic version of this very versatile technique.

Although this technique has a very benign-looking formula for Euclidean spaces, the complexity of the same procedure increases many times when one formulates it over a non-Euclidean space. This happens because one has to now start using the various differential geometric constructs which typically do not need explicit formulations in the Euclidean space. We will assume a basic familiarity with these differential geometric constructs, only defining the narrow set of concepts that we will be using directly in our algorithms and results. We do not aim to provide a comprehensive understanding of any of these concepts in this thesis. Look at the excellent manifold books [71–73] by John M. Lee or any other standard references for the background.

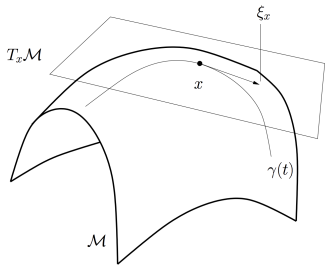


Figure 1.3: Geodesic on a manifold.

Algorithm 1.1 RiemannianGD$(F, x^{(0)}, N, t)$

$x^{(0)}$ is the starting point for GD in $\mathcal{M}$
N is the number of iterations and t is the step size
for $i \in [N]$ **do**
    $x^{(i)} \leftarrow \gamma_{x^{(i-1)}}\left(-\nabla F_{x^{(i-1)}}, t\right)$ */* Gradient descent step */*
Return: $x^{(N)}$

To describe the gradient descent procedure over non-Euclidean spaces, we need to describe the procedure using the geometry of the space. Specifically, we will need two quantities: Riemannian gradients and geodesics. Riemannian gradients play the same role as that of gradients in the Euclidean space (they provide the direction in which to move) while geodesics being the equivalent of a straight line in the Euclidean space provide the actual path to move along.

We demonstrate this in Figure 1.3. Given a point $x \in \mathcal{M}$ and a tangent vector $\xi_x \in T_x\mathcal{M}$ (direction of descent in our case), $\gamma : [0, t) \to \mathcal{M}$ is a curve s.t. $\gamma(0) = x$ and $\gamma'(0) = \xi_x$. We use the notation $\gamma_x(\xi_x, t)$ to denote the geodesic at the point x and moving in the direction ξ_x. With this notation, the Riemannian gradient descent algorithm is given in Algorithm 1.1.

In Chapters 3 and 4, we will model Euclidean optimization as optimization problems over manifolds. Here we will need explicit formulae to compute Riemannian gradients and geodesics, which we will provide and restate the detailed gradient descent in their terms.

Optimization problems over manifolds have been studied in the past for solving various problems. These problems tend to exploit the fact that the set of desirable solutions forms the underlying manifold and hence instead of defining a function in the Euclidean space by mapping the points on the manifold to the Euclidean space, one can directly optimize over the desired manifold. An example of this is the Rayleigh quotient problem [3].

Our use of the gradient descent procedure over the manifold differs considerably from this usual approach. We take a problem in the Euclidean space whose desired solution is also a point in the same Euclidean space but we use the manifold to break the problem down into small parts. Each of these small parts can be solved on their own and since the set of all of these small parts forms a manifold we can use an optimization procedure over the manifold to get the final answer for the full problem.

1.1.3 Minimax Estimation and Statistical Games

Let $\mathcal{P} = \{P_\theta : \theta \in \Theta \subseteq \mathbb{R}^d\}$ be a parametric family of distributions. In this work, we assume Θ is a compact set. Let $\mathbb{X}^n = \{X_1, \dots X_n\} \in \mathcal{X}^n$ be n independent samples drawn from some unknown distribution $P_\theta \in \mathcal{P}$. Given $\mathbb{X}^n$, our goal is to estimate the unknown parameter θ. A deterministic estimator $\hat{\theta}$ of θ is any measurable function from $\mathcal{X}^n$ to Θ. We denote the set of deterministic estimators by $\mathcal{D}$. A randomized estimator is given by a probability measure on the set of deterministic estimators. Given $\mathbb{X}^n$, the unknown parameter θ is estimated by first sampling a deterministic estimator according to this probability measure and using the sampled estimator to predict θ. Since any randomized estimator can be identified by a probability measure on $\mathcal{D}$, we denote the set of randomized estimators by $\mathcal{M}_\mathcal{D}$, the set of all probability measures on $\mathcal{D}$. Let $M : \Theta \times \Theta \to \mathbb{R}$ be a measurable loss function such that $M(\theta', \theta)$ measures the cost of an estimate θ' when the true parameter is θ. Define the risk of an estimator $\hat{\theta}$ for estimating θ as $R(\hat{\theta}, \theta) \stackrel{\text{def}}{=} \mathbb{E}\left[M(\hat{\theta}(\mathbb{X}^n), \theta)\right]$, where the expectation is taken with respect to randomness from $\mathbb{X}^n$ and the estimator $\hat{\theta}$. The worst-case risk of an estimator $\hat{\theta}$ is defined as $\sup_{\theta \in \Theta} R(\hat{\theta}, \theta)$ and the minimax risk is defined as the best worst-case risk that can be achieved by any estimator

$$R^* \stackrel{\text{def}}{=} \inf_{\hat{\theta} \in \mathcal{M}_\mathcal{D}} \sup_{\theta \in \Theta} R(\hat{\theta}, \theta). \tag{1.1}$$

Any estimator whose worst case risk is equal to the minimax risk is called a minimax estimator. We refer to the above min-max problem as a *statistical game*. Often, we are also interested in deterministic minimax estimators, which are defined as estimators with worst case risk equal to

$$\inf_{\hat{\theta} \in \mathcal{D}} \sup_{\theta \in \Theta} R(\hat{\theta}, \theta). \tag{1.2}$$

From the perspective of game theory, the optimality notion in Equation (1.1) is referred to as the *minmax* value of the game. This is to be contrasted with the *maxmin* value of the game $\sup_{\theta \in \Theta} \inf_{\hat{\theta} \in \mathcal{M}_\mathcal{D}} R(\hat{\theta}, \theta)$. In general, these two quantities are **not** equal, but the following relationship always holds:

$$\sup_{\theta \in \Theta} \inf_{\hat{\theta} \in \mathcal{M}_\mathcal{D}} R(\hat{\theta}, \theta) \leq \inf_{\hat{\theta} \in \mathcal{M}_\mathcal{D}} \sup_{\theta \in \Theta} R(\hat{\theta}, \theta). \tag{1.3}$$

In statistical games, for typical choices of loss functions, $\sup_{\theta\in\Theta}\inf_{\hat{\theta}\in\mathcal{M}_{\mathcal{D}}} R(\hat{\theta},\theta) = 0$, whereas $\inf_{\hat{\theta}\in\mathcal{M}_{\mathcal{D}}}\sup_{\theta\in\Theta} R(\hat{\theta},\theta) > 0$; that is, the minmax value is strictly greater than maxmin value of the game. So we cannot in general reduce computing the minmax value to computing the maxmin value.

Linearized Statistical Games. Without any additional structure such as convexity, computing the values of min-max games is difficult in general. So it is common in game theory to consider a *linearized game* in the space of probability measures, which is in general better-behaved. To set up some notation, for any probability distribution P, define $R(\hat{\theta}, P)$ as $\mathbb{E}_{\theta\sim P}\left[R(\hat{\theta},\theta)\right]$. In the context of statistical games, a linearized game has the following form:

$$\inf_{\hat{\theta}\in\mathcal{M}_{\mathcal{D}}}\sup_{P\in\mathcal{M}_{\Theta}} R(\hat{\theta}, P), \tag{1.4}$$

where $\mathcal{M}_{\Theta}$ is the set of all probability measures on Θ. The minmax and maxmin values of the linearized game and the original game in Equation (1.1) are related as follows

$$\sup_{\theta\in\Theta}\inf_{\hat{\theta}\in\mathcal{M}_{\mathcal{D}}} R(\hat{\theta},\theta) \le \sup_{P\in\mathcal{M}_{\Theta}}\inf_{\hat{\theta}\in\mathcal{M}_{\mathcal{D}}} R(\hat{\theta}, P) \le \inf_{\hat{\theta}\in\mathcal{M}_{\mathcal{D}}}\sup_{P\in\mathcal{M}_{\Theta}} R(\hat{\theta}, P) \overset{(a)}{=} \inf_{\hat{\theta}\in\mathcal{M}_{\mathcal{D}}}\sup_{\theta\in\Theta} R(\hat{\theta},\theta),$$

where (a) holds because for any estimator $\hat{\theta}$, $\sup_{P\in\mathcal{M}_{\Theta}} R(\hat{\theta}, P)$ is equal to $\sup_{\theta\in\Theta} R(\hat{\theta},\theta)$. Thus, the minmax values of the original and linearized statistical games are equal. Any estimator whose worst-case risk is equal to the minmax value of the linearized game is a minimax estimator. The maxmin values of the original and linearized statistical games are however in general different. In particular, as discussed above, the maxmin value of the original statistical game is usually equal to zero. The maxmin value of the *linearized game* however has a deep connection to Bayesian estimation.

Note that $R(\hat{\theta}, P)$ is simply the integrated risk of the estimator $\hat{\theta}$ under prior $P \in \mathcal{M}_{\Theta}$. Any estimator which minimizes $R(\hat{\theta}, P)$ is called the Bayes estimator for P, and the corresponding minimum value is called Bayes risk. Though the set of all possible measurable estimators is in general vast, in what might be surprising from an optimization or game-theoretic viewpoint, the Bayes estimator can be characterized simply as follows. Letting $P(\cdot|\mathbb{X}^n)$ be the posterior distribution of θ given the data $\mathbb{X}^n$, a Bayes estimator of P can be found by minimizing the posterior risk

$$\hat{\theta}_P(\mathbb{X}^n) \in \operatorname*{argmin}_{\tilde{\theta}\in\Theta} \mathbb{E}_{\theta\sim P(\cdot|\mathbb{X}^n)}\left[M(\tilde{\theta},\theta)\right]. \tag{1.5}$$

Certain mild technical conditions need to hold for $\hat{\theta}_P$ to be measurable and for it to be a Bayes estimator [9]. We detail these conditions in Section 5.8.1, which incidentally are all satisfied for the problems considered in this work. A least favourable prior is defined as any prior which maximizes the Bayes risk; that is, $\tilde{P}$ is LFP if $\inf_{\hat{\theta}\in\mathcal{M}_{\mathcal{D}}} R(\hat{\theta}, \tilde{P}) = \sup_{P\in\mathcal{M}_{\Theta}}\inf_{\hat{\theta}\in\mathcal{M}_{\mathcal{D}}} R(\hat{\theta}, P)$. Thus, LFPs solve for the maxmin value of the linearized statistical game. Any prior whose Bayes risk is equal to the maxmin value of the linearized game is an LFP.

Nash Equilibrium. Directly solving for the minmax or maxmin values of the (linearized) min-max games is in general computationally hard, in large part because: (a) these values need not be equal, which limits the set of possible optimization algorithms, and (b)

the optimal solutions need not be stable, which makes it difficult for simple optimization problems. It is thus preferable that the two values are equal[1], and the solutions be stable, which is formalized by the game-theoretic notion of a *Nash equilibrium* (NE).

For the original statistical game in Equation (1.1), a pair $(\hat{\theta}^*, \theta^*) \in \mathcal{M}_{\mathcal{D}} \times \Theta$ is called a pure strategy NE, if the following holds

$$\sup_{\theta \in \Theta} R(\hat{\theta}^*, \theta) \leq R(\hat{\theta}^*, \theta^*) \leq \inf_{\hat{\theta} \in \mathcal{M}_{\mathcal{D}}} R(\hat{\theta}, \theta^*) = \inf_{\hat{\theta} \in \mathcal{D}} R(\hat{\theta}, \theta^*),$$

where the equality follows since the optimum of a linear program over a convex hull can always be attained at an extreme point. Intuitively, this says that there is no incentive for any player to change their strategy while the other player keeps hers unchanged. Note that whenever a pure strategy NE exists, the minmax and maxmin values of the game are equal to each other:

$$\inf_{\hat{\theta} \in \mathcal{M}_{\mathcal{D}}} \sup_{\theta \in \Theta} R(\hat{\theta}, \theta) \leq \sup_{\theta \in \Theta} R(\hat{\theta}^*, \theta) \leq R(\hat{\theta}^*, \theta^*) \leq \inf_{\hat{\theta} \in \mathcal{M}_{\mathcal{D}}} R(\hat{\theta}, \theta^*) \leq \sup_{\theta \in \Theta} \inf_{\hat{\theta} \in \mathcal{M}_{\mathcal{D}}} R(\hat{\theta}, \theta).$$

Since the RHS is always upper bounded by the LHS from (1.3), the inequalities above are all equalities.

As we discussed above, the maxmin and minmax values of the statistical game in Equation (1.1) are in general not equal to each other, so that a pure strategy NE will typically not exist for the statistical game (1.1). Instead what often exists is a mixed strategy NE, which is precisely a pure strategy NE of the linearized game. That is, $(\hat{\theta}^*, P^*) \in \mathcal{M}_{\mathcal{D}} \times \mathcal{M}_{\Theta}$ is called a mixed strategy NE of statistical game (1.1), if

$$\sup_{\theta \in \Theta} R(\hat{\theta}^*, \theta) = \sup_{P \in \mathcal{M}_{\Theta}} R(\hat{\theta}^*, \theta) \leq R(\hat{\theta}^*, P^*) \leq \inf_{\hat{\theta} \in \mathcal{M}_{\mathcal{D}}} R(\hat{\theta}, P^*) = \inf_{\hat{\theta} \in \mathcal{D}} R(\hat{\theta}, P^*).$$

As with the original game, if $(\hat{\theta}^*, P^*)$ is a pure strategy NE of the linearized game of (1.1), aka, a mixed strategy NE of (1.1), then the minmax and maxmin values of the linearized game are equal to each other, and, moreover $\hat{\theta}^*$ is a minimax estimator and P^* is an LFP. Conversely, if $\hat{\theta}^*$ is a minimax estimator, and P^* is an LFP, and the minmax and maxmin values of (1.4) are equal to each other, then $(\hat{\theta}^*, P^*)$ is a mixed strategy NE of (1.1). These just follow from similar sandwich arguments as with the original game, which we add for completeness in Section 5.8.2.

In gist, it might be computationally easier to recover the mixed strategy NE of the statistical game, assuming they exist, and doing so, would recover minimax estimators and LFPs. In this work, we are thus interested in imposing mild conditions on the statistical game so that a mixed strategy NE exists, and under this setting, develop tractable algorithms to estimate the mixed strategy NE.

Existence of NE. We now briefly discuss sufficient conditions for the existence of NE. As discussed earlier, a pure strategy NE does not exist for statistical games in general. So, here we focus on existence of mixed strategy NE. In a seminal work, Wald [104] studied the conditions for existence of a mixed strategy NE, and showed that a broad class of

[1] John Von Neumann, a founder of game theory, has said he could not foresee there even being a theory of games without a theorem that equates these two values

statistical games have mixed strategy NE. Suppose every distribution in the model class $\mathcal{P}$ is absolutely continuous, Θ is compact, and the loss M is a bounded, non-negative function. Then minmax and maxmin values of the linearized game are equal. Moreover, a minimax estimator with worst-case risk equal to R^* exists. Under the additional condition of compactness of $\mathcal{P}$, [104] showed that an LFP exists as well. Thus, based on our previous discussion, this implies the game has a mixed strategy NE. In this work, we consider a different and simpler set of conditions on the statistical game. We assume that Θ is compact and the risk $R(\hat{\theta}, \theta)$ is Lipschitz in its second argument. Under these assumptions, we show that the minmax and maxmin values of the linearized game in Equation (1.4) are equal to each other. Such results are known as minimax theorems and have been studied in the past [103, 104, 110]. However, unlike past works that rely on fixed point theorems, we rely on a constructive learning-style proof to prove the minimax theorem, where we present an algorithm which outputs an approximate NE of the statistical game. Under the additional condition that the risk $R(\hat{\theta}, \theta)$ is bounded, we show that the statistical game has a minimax estimator and an LFP.

Computation of NE. Next, we discuss previous numerical optimization techniques for computing a mixed strategy NE of the statistical game. Note that this is a difficult computational problem: minimizing over the domain of all possible estimators, and maximizing over the set of all probability measures on Θ. Nonetheless, several works in statistics have attempted to tackle this problem [9]. One class of techniques involves reducing the set of estimators $\mathcal{D}$ via admissibility considerations to a small enough set. Given this restricted set of estimators, they can then directly calculate a minimax test for some testing problems; see for instance Hald [45]. A drawback of these approaches is that they are restricted to simple estimation problems for which the set of admissible estimators are easy to construct. Another class of techniques for constructing minimax estimators relies on the properties of LFPs [26, 55]. When the parameter set Θ is a compact subset of $\mathbb{R}$, and when certain regularity conditions hold, it is well known that LFPs are supported on a finite set of points [9, 40]. Based on this result, Kempthorne [59], Nelson [86] propose numerical approaches to determine the support points of LFPs and the probability mass that needs to be placed on these points. However, these approaches are restricted to 1-dimensional estimation problems and are not broadly applicable. In a recent work, Luedtke et al. [77] propose heuristic approaches for solving statistical games using deep learning techniques. In particular, they use neural networks to parameterize the statistical game and solve the resulting game using local search techniques such as alternating gradient descent. However, these approaches are not guaranteed to find minimax estimators and LFPs and can lead to undesirable equilibrium points. They moreover parameterize estimators via neural networks whose inputs are a simple concatenation of all the samples, which is not feasible for large n.

In our work, we develop numerical optimization techniques that rely on online learning algorithms (see Section 1.1.4). Though the domains as well as the setting of the statistical game are far more challenging than typically considered in learning and games literature, we reduce the problem of designing minimax estimators to a purely computational problem of efficient implementation of certain optimization subroutines. For the wide range of problems where these subroutines can be efficiently implemented, our algorithm provides an efficient and scalable technique for constructing minimax estimators.

1.1.4 Online Learning

The online learning framework can be seen as a repeated game between a learner/decision-maker and an adversary. In this framework, in each round t, the learner makes a prediction $\mathbf{x}_t \in \mathcal{X}$, where $\mathcal{X} \subseteq \mathbb{R}^d$, and the adversary chooses a loss function $f_t : \mathcal{X} \to \mathbb{R}$ and observe each others actions. The goal of the learner is to choose a sequence of actions $\{\mathbf{x}_t\}_{t=1}^T$ so that the cumulative loss $\sum_{t=1}^T f_t(\mathbf{x}_t)$ is minimized. The benchmark with which the cumulative loss will be compared is called the best fixed policy in hindsight, which is given by $\inf_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^T f_t(\mathbf{x})$. This results in the following notion of regret, which the learner aims to minimize

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \inf_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^{T} f_t(\mathbf{x}).$$

When the domain $\mathcal{X}$ is compact, and convex, and the loss functions f_t are convex: Under this simple setting, a number of efficient algorithms for regret minimization have been studied. Some of these include Follow the Regularized Leader (FTRL) [48, 81], Follow the Perturbed Leader (FTPL) [57]. In FTRL, one predicts $\mathbf{x}_t$ as $\operatorname{argmin}_{\mathbf{x}\in\mathcal{X}} \sum_{i=1}^{t-1} f_i(\mathbf{x}) + r(\mathbf{x})$, where r is a strongly convex regularizer. In FTPL, one predicts $\mathbf{x}_t$ as $\mathbb{E}_\sigma \left[\operatorname{argmin}_{\mathbf{x}\in\mathcal{X}} \sum_{i=1}^{t-1} f_i(\mathbf{x}) - \langle\sigma, \mathbf{x}\rangle\right]$, where σ is a random perturbation drawn from some appropriate probability distribution such as exponential distribution. These algorithms are known to achieve the optimal $O(\sqrt{T})$ regret in the convex setting [81, 96].

When $\mathcal{X}$ is compact, but either the domain or the loss functions f_t are non–convex: Under this setting, no deterministic algorithm can achieve sub-linear regret (i.e., regret which grows slower than T) [21, 96]. In such cases one has to rely on randomized algorithms to achieve sub-linear regret. In randomized algorithms, in each round t, the learner samples the prediction $\mathbf{x}_t$ from a distribution $P_t \in \mathcal{M}_\mathcal{X}$, where $\mathcal{M}_\mathcal{X}$ is the set of all probability distributions supported on $\mathcal{X}$. The goal of the learner is to choose a sequence of distributions $\{P_t\}_{t=1}^T$ to minimize the expected regret $\sum_{t=1}^T \mathbb{E}_{\mathbf{x}\sim P_t}[f_t(\mathbf{x})] - \inf_{\mathbf{x}\in\mathcal{X}} \sum_{t=1}^T f_t(\mathbf{x})$. An alternative perspective of such randomized algorithms is as deterministic algorithms solving a *linearized problem* in the space of probability distributions, with loss functions $\tilde{f}_t(P) = \mathbb{E}_{\mathbf{x}\sim P}[f_t(\mathbf{x})]$, and rely on algorithms for online convex learning. For example, by relying of FTRL, one predicts P_t as

$\operatorname{argmin}_{P\in\mathcal{M}_\mathcal{X}} \sum_{i=1}^{t-1} \tilde{f}_i(P) + r(P)$, for some strongly convex regularizer $r(P)$. When $r(P)$ is the negative entropy of P, Krichene et al. [67] show that the resulting algorithm achieves $O(\sqrt{dT \log T})$ expected regret.

Another technique to solve the linearized problem is via the FTPL algorithm [4, 96]. In this algorithm, P_t is given by the distribution of the random variable $\mathbf{x}_t(\sigma)$, which is a minimizer of $\sum_{i=1}^{t-1} f_i(\mathbf{x}) - \langle\sigma, \mathbf{x}\rangle$. Here, σ is a random perturbation drawn from some appropriate probability distribution. In recent work, Suggala and Netrapalli [96] show that this algorithm achieves $O(\sqrt{d^3 T})$ expected regret.

Without any assumptions on $\mathcal{X}$ or the loss functions f_t. A key caveat with statistical games is that the domain of all possible measurable estimators is not bounded and is an infinite-dimensional space. Thus, results as discussed above from the learning and games literature are not applicable to such a setting. In particular, regret bounds of FTRL and FTPL scale with the dimensionality of the domain, which is infinite in this case. But

there is a very simple strategy that is applicable without making any assumptions on the domain whatsoever, but under the provision that f_t *was known* to the learner ahead of round t. Then, an optimal strategy for the learner is to predict $\mathbf{x}_t$ as simply a minimizer of $f_t(\mathbf{x})$. It is easy to see that this algorithm, known as Best Response (BR), has 0 regret. While this is an impractical algorithm in the framework of online learning, it can be used to solve min-max games, as we will see in Section 5.1.

FTPL. We will be making use of the FTPL algorithm in the sequel, so we now describe this in a bit more detail. In this algorithm, the learner predicts $\mathbf{x}_t$ as a minimizer of $\sum_{i=1}^{t-1} f_i(\mathbf{x}) - \langle \sigma, \mathbf{x} \rangle$, where $\sigma \in \mathbb{R}^d$ is a random perturbation such that $\{\sigma_j\}_{j=1}^d \overset{i.i.d}{\sim} \text{Exp}(\eta)$ and $\text{Exp}(\eta)$ is the exponential distribution with parameter η[2]. When the domain $\mathcal{X}$ is bounded and loss functions $\{f_t\}_{t=1}^T$ are Lipschitz (not necessarily convex), FTPL achieves $O(\sqrt{d^3 T})$ expected regret, for appropriate choice of η [96]. A similar regret bound holds even when $\mathbf{x}_t$ is an approximate minimizer of $\sum_{i=1}^{t-1} f_i(\mathbf{x}) - \langle \sigma, \mathbf{x} \rangle$. Suppose for any $t \in \mathbb{N}$, $\mathbf{x}_t$ is such that

$$\sum_{i=1}^{t-1} f_i(\mathbf{x}_t) - \langle \sigma, \mathbf{x}_t \rangle \leq \inf_{\mathbf{x} \in \mathcal{X}} \sum_{i=1}^{t-1} f_i(\mathbf{x}) - \langle \sigma, \mathbf{x} \rangle + (\alpha + \beta \|\sigma\|_1),$$

where α, β are positive constants. Then FTPL achieves $O(T^{1/2} + \alpha T + \beta T^{3/2})$ expected regret for appropriate choice of η (see Section 5.8.3 for more details).

[2]Recall, X is an exponential random variable with parameter η if $P(X \geq s) = \exp(-\eta s)$

Part I

Data poisoning in Machine Learning

2 | A new stochastic optimization technique for combating data poisoning attacks

We study methods for stochastic optimization in the setting where a subset of training data might be corrupted by an adversary. Stochastic methods in optimization have become an important workhorse in the practice of modern Machine Learning. These methods usually work on data collected from a variety of different sources (such as scraping the Internet). Naturally, an adversary can inject malicious data in this training set and it might be very challenging to detect and remove this corrupted subset. In such a scenario, it is desirable to have algorithms which are immune to injections of small amount of arbitrary corruptions. In this paper, we propose a novel method for stochastic optimization which has the potential of addressing this problem for a wide class of loss functions.

Difficulty of the problem. The problem of learning good models under worst case corruptions in training data is NP-hard even for simple problems like binary classification with half-spaces [44]. The popular method of dealing with these difficulties is to make distributional assumptions over the data or the noise added by the adversary [61]. These distributional assumptions heavily dictate the design of appropriate algorithms in these settings. However, one doesn't always know how the distributions will look like in practice and, moreover, in the case of noise a determined attacker might tailor it to the specific training data at hand and hence invalidate any distributional assumptions made. This makes the problem of building optimization problems which are robust to worst-case noise in the training data seemingly intractable.

In this work, we propose a tractable way out of this difficulty. Instead of expecting our algorithm to behave perfectly on *all* input instances and be able to handle *all* worst-case noise (which makes the problem NP-hard), we build an algorithm that performs well on *most* instances and handles worst-case noise on these instances without putting any distributional assumptions on either the training data or the noise. By working well on *most* input instances we expect to capture all the instances that one could reasonably expect to see in practice, while leaving out the small fraction of instances which are often responsible for the computational hardness of a given problem (for example, the ones to which a reduction from an NP-hard problem like satisfiability might map to).

The idea is to develop an algorithm whose output does not change drastically under small

perturbations in the training data. This algorithm is not designed to necessarily perform well on a small set of instances, even when some other algorithm might be able to solve these instances well. The hope is that this small set of instances is one that will not be seen in practice. Since characterizing the complexity landscape of various instances is a highly intricate and mathematically extremely challenging subject [5], the above discussion is meant to only give an intuitive understanding of our approach. As we shall see later in Section 2.5, we achieve our objective by identifying a crucial property (called the *gap parameter*) of the solutions of a given optimization problem. Our algorithm will only find solutions that have a large enough gap parameter, giving them good robustness properties under perturbations in the input data.

Our approach. Instead of working in the original dimension of the given optimization problem, our algorithm proceeds by solving the given problem in a sequence of random hyperplanes of a smaller dimension. These hyperplanes are defined so that each successive one contains the solution obtained from the previous one. The algorithm stops either when the improvements in the successive hyperplanes become small enough or after a chosen number of iterations, and outputs the solution obtained in the last hyperplane. See Algorithms 2.1 and 2.2.

Significance of our approach. Our approach to stochastic optimization algorithms has interesting connections with the theory of expander graphs. The role of the gap parameter in our analysis is akin to that of the spectral gap of the Laplacian of a graph. Under very mild assumptions on this parameter (the logarithm of this parameter has to be polynomially bounded in the dimension of the problem) we obtain a polynomially bounded runtime for our algorithm (see Theorem 2.1).

The second eigenvalue is an important spectral quantity for graphs [66] and our analysis shows that a similar quantity for functions dictates how fast certain stochastic optimization algorithms can converge. This is distinct from all previous analyses which rely on structural properties of the functions like convexity, Lipschitzness, or the Polyak-Lojasiewicz criterion to bound the rate of convergence. As far as we know, such a parallel between the very well-studied theory of random walks on expander graphs and stochastic optimization algorithms is entirely novel to our work.

The highlight of our analysis is Lemma 2.2, which is a statement about moving non-trivially away from the maximum of a function by random sampling. For a function whose domain is a Lie group which satisfies Kazhdan's Property (T), it gives a lower bound on the size of the set where the function takes a value non-trivially away from the maximum. This is a very general lemma (see Section 2.4) and we believe that it will find applicability in many future analyses.

Organization of this paper. In Section 2.1, we discuss existing approaches for stochastic optimization and dealing with perturbations in training data, popularly referred to as data poisoning attacks. In Section 2.2, we set up the notation and introduce all the concepts needed for the rest of the paper. In Section 2.3, we describe and discuss our stochastic optimization algorithm and give our convergence result. In Section 2.4, we discuss the main theoretical technique of our analysis. In Section 2.5, we discuss the robustness of our algorithm and demonstrate the efficacy of our approach with experiments in Section 2.6.

2.1 Related Work

In this section, we compare our approach to stochastic optimization with existing approaches as well as discuss the literature on data poisoning attacks in Machine Learning.

Comparison to existing stochastic techniques. Most of the existing literature focuses on either stochastic gradient descent or it's popular variants like Adam [63] and AdaGrad [30]. In stochastic gradient descent one picks a random subset of the data, computes the loss on this subset and uses the gradient of this loss to update the parameters of the model. Convergence for this scheme can be shown under assumption like strong convexity [84], the Polyak-Lojasiewicz condition [41], and convergence to stationary points for non-convex functions which satisfy an expected smoothness assumption [60]. These convergence results rely crucially on the respective structural properties mentioned for the loss functions, while the randomness of picking a subset of the data usually worsens the convergence rates as compared to their deterministic counterparts (which work with the full training data in all iterations).

Our approach is fundamentally different from these approaches. Instead of subsets of the training data being the source of randomness, in our approach the randomness comes from the selection of random subspaces in which the given optimization problem is solved. In the particular case when the optimization problem is solving for optimal parameters in a Euclidean space, our method works in subspaces of the full space of the parameters. The only existing technique that has superficial similarities to this is the dropout method in deep learning [95]. But even there one typically considers only subsets and not subspaces of the parameters. Note that the set of all subspaces of the parameter space is a much bigger space (being a smooth manifold) than the set of all of their subsets (which is a discrete set). In addition, dropout is a specialized technique that is only used in the context of deep learning.

Analytically, our analysis is dependent on the crucial fact that the space our randomness is drawn from forms a smooth manifold that it is a quotient of a compact Lie group, and in particular therefore satisfies Kazhdan's Property (T). The only assumption we need from the loss function is that it should be smooth. We do not need any other assumptions like convexity or Lipschitzness.

Data poisoning in Machine Learning. Many methods exist in the literature for dealing with data poisoning; see [98] and [25] for excellent surveys. While there are a lot of methods which try to deal with data poisoning for specific models like linear regression, logistic regression, or neural networks, few methods exist which have general applicability. Data sanitization and some form of bagging and majority voting seem to be among these few general techniques. Data sanitization can be difficult as adversaries assemble more and more sophisticated forms of noise to make noisy data look indistinguishable from real data. Bagging and voting can decrease the amount of data available for training for a single model and can have unwanted accuracy trade-offs. Robust training, which augments the training data with poisoned instances to specifically train the model to handle such data, is another popular method to combat such attacks. All of these techniques deal with the preprocessing of the training data, and not with the actual learning process.

Techniques like [89] and [22], which deal with the learning process, have been developed

in the robust statistics literature to mitigate the influence of noise in the training data. But these techniques tend to be intractable without making restrictive distributional or modeling assumptions.

Our technique, which is primarily a new optimization algorithm, can be used either as an alternative to these existing techniques or in conjunction with them to provide enhanced protection against data poisoning attacks.

2.2 Preliminaries

Notation. We use G to represent a Lie group and H to represent a subgroup of it. Moreover, we use G/H to represent the quotient of G w.r.t. H. For a treatment of Lie groups see [17]. We use $O(d)$ to represent the compact orthogonal group acting on $\mathbb{R}^d$. The product group $O(k) \times O(d-k)$ can naturally be identified as a subgroup of $O(d)$. The quotient $O(d)/\left(O(k) \times O(d-k)\right)$ has a natural interpretation as the set of all k-dimensional subspaces of $\mathbb{R}^d$. This is a well studied geometric object, popularly known as the Grassmannian (see [8]). We denote it by $G_{k,d}$. We use the term k-plane to refer to a k-dimensional affine subspace, i.e. a k-dimensional hyperplane of $\mathbb{R}^d$, in the rest of the paper.

For us, $\ell : \mathbb{R}^d \to \mathbb{R}$ will be the smooth loss function we want to optimize. Here, smoothness means that ℓ is infinitely differentiable. We use η with various subscripts to represent subspaces or k-planes of appropriate dimensions (which will be clear from the context).

Measures on Lie Groups. Our Lie groups, like all locally compact Lie groups, have a left-invariant Haar measure which is unique up to scaling [85]. This covers a wide range of Lie groups used in applications [38]. For results regarding existence of invariant measures on compact Lie groups, their quotients (like the Grassmannians) and validity of Fubini style decompositions look at Chapter 1 of [94].

For a measurable subset A of a given measure space we use $|A|$ to denote the measure of this set under the implied measure.

Kazhdan's Property (T). For a definition of this property see Section 3.1 of [91]. It is primarily defined for non-compact Lie groups. Indeed for compact Lie groups, such as we are considering in this paper, the property is trivially satisfied. We bring it up here because of its impact on expander graphs and their random walks which forms an important motivation for our work. Also, because we formulate our core lemma, Lemma 2.2, on non-compact Lie groups. We will only be working with the following consequence of the property in our proofs:

Lemma 2.1. *[Remark 1.1.4 in [7]] Let G be a locally compact Lie group that satisfies Kazhdan's property (T). Then there exists a $c > 0$ such that for all functions $f : G \to \mathbb{R}$, square integrable w.r.t. a left-invariant Haar measure and which satisfy $\int_G f = 0$, there exists a $\gamma \in G$ satisfying*

$$\|f - \gamma \cdot f\|^2 \geq c\|f\|^2$$

where the action of γ on f is defined by $(\gamma \cdot f)(x) = f(\gamma^{-1} \cdot x)$.

Algorithm 2.1 Our Random Walk

Input: $\ell : \mathbb{R}^d \to \mathbb{R}, x_0 \in \mathbb{R}^d, d > k > 1, T \geq 0, N \geq 0$

1: **for** $i = 1, \ldots, N$ **do**
2: Sample $\eta_1, \ldots, \eta_T$ uniformly from $G_{k,d}$
3: $y_j \leftarrow \arg\min_{y \in x_{i-1}+\eta_j} \ell(y)$ for $j \in [T]$
4: $x_i \leftarrow \operatorname{argmin}_{y \in \{y_1, \ldots, y_T\}} \ell(y)$
5: **return** x_N

Remark 2.1. *The constant $c > 0$ in Lemma 2.1 is only dependent on the group and is popularly referred to as the Kazhdan constant of the group. We can chose $c = 2$ for compact Lie groups. The proof of Lemma 2.2 in Appendix 2.7.2 includes a proof of this fact.*

Noise model. We will study the robustness properties of our techniques in Section 2.5. We consider noise only in the training data matrix A. Noise may be introduced by perturbing a certain fraction of the rows of A with a noise matrix Δ or by augmenting A with a small number of well crafted data points. Generally, both these settings can be mathematically modelled as adding noise Δ to A. This setting is popularly referred to as *data poisoning*. We study the behavior of our approach as the fraction of rows that Δ corrupts increases. Note that we do not make any distributional assumptions on Δ, instead we work with the worst case Δ by evaluating our technique against existing data poisoning attacks in the literature, which generate Δ with full knowledge of A.

2.3 Our Results

In this section, we describe our random walk, which is a stochastic optimization technique applicable to any smooth loss function. Moreover, we provide a convergence result that works in this very general setting.

2.3.1 Random Walk

The aim of any optimization algorithm is to find some critical point of ℓ, usually one of the global minima, i.e., find an $x^* \in \mathbb{R}^d$ such that

$$x^* \in \arg\min_{x \in \mathbb{R}^d} \ell(x)$$

Assume that we are given a black-box access for solving the same problem but in a smaller-dimensional space, specifically a k-plane $\eta \subset \mathbb{R}^d$, i.e., we can find an x^*_η such that

$$x^*_\eta \in \arg\min_{x \in \eta} \ell(x)$$

Our random walk is motivated by asking the question: Can we use this black box repeatedly for a sequence of k-planes $\eta_1, \eta_2, \ldots$ to find an x^*? This suggests a natural random walk as follows: start with some $x_1 \in \mathbb{R}^d$ and sample a random k-plane η_1 containing x_1; find an x_2 such that $x_2 \in \arg\min_{x \in \eta_1} \ell(x)$; in i-th step find a random k-plane η_i containing x_i and solve for $\arg\min_{x \in \eta_i} \ell(x)$; stop the algorithm after N steps. We state this more formally in Algorithm 2.1.

This is a very natural random walk from computational complexity theory perspective. It leverages the ability to solve several smaller-dimensional random problems when solving a bigger-dimensional problem. This approach has been used to study other problems in the literature (see Section 10.1.2 in [5]), and has provided interesting insights in to the structure of these problems. This approach is called random self-reducibility. To the best of our knowledge, our work is the first to study this approach for optimization problems in the Euclidean space.

One of the surprising observations from the practice of modern ML is the ease of solving many seemingly intractable non-convex optimization problems. This justifies the use of a black-box to solve a problem of a smaller dimension in Algorithm 2.1. Since our random walk is designed to serve the dual purpose of optimization as well as learning a model robust to data poisoning, one would not be advised to use the same black-box to solve the original problem directly. The black box can be implemented with any of the existing techniques. We recommend using a technique best suited to the specific machine learning model that is being learned. We now study the convergence properties of Algorithm 2.1.

2.3.2 Convergence Analysis

For the convergence analysis, we need to define a few auxiliary functions. We define $L : \mathbb{R}^d \times G_{k,d} \to \mathbb{R}, M : \mathbb{R}^d \to \mathbb{R}, m : \mathbb{R}^d \to \mathbb{R}, \Theta : \mathbb{R}^d \to \mathbb{R}$ and $\theta : \mathbb{R} \to \mathbb{R}$ as follows:

$$\begin{aligned} L(x,\eta) &\coloneqq \min_{y \in x+\eta} \ell(y), \\ M(x) \coloneqq \max_{\eta \in G_{k,d}} L(x,\eta), \quad m(x) &\coloneqq \min_{\eta \in G_{k,d}} L(x,\eta), \\ \Theta(x) \coloneqq \frac{\|L(x,\cdot)\|_2^2}{2|M(x)-m(x)|^2}, \quad \theta(\alpha) &\coloneqq \min_{x \in \{x : \ell(x) = \alpha\}} \Theta(x) \end{aligned}$$

We call θ the **gap function** of ℓ and $\theta(\ell(x))$ the **gap parameter** of the minimizer x. The gap function of ℓ plays a crucial role in our analysis and have very close connections with the spectral gap of a Laplacian on a graph. We discuss this connection in more detail in the next section. Our main convergence proof is as follows:

Theorem 2.1. *Let $\ell : \mathbb{R}^d \to \mathbb{R}$ be a smooth loss function such that $\theta(\ell) \geq 1 - \delta$ for some $\delta > 0$. Let $\alpha = \min_x \ell(x)$. For all ϵ_0 and γ in $(0,1)$, with $N = \frac{\log 1/\epsilon_0}{\log 2/\delta}$ and $T = \frac{\log N + \log 2/3\gamma}{\log 1/\delta}$ and with probability at least $1-\gamma$, Algorithm 2.1 finds an $x \in \mathbb{R}^d$ such that*

$$\ell(x) - \alpha \leq \epsilon_0(\ell(x_0) - \alpha).$$

We defer the proof of Theorem 2.1 to Appendix 2.7.5 and discuss the main theoretical ideas behind it in Section 2.4. For now, there are several interesting points to note about Theorem 2.1:

1. It only uses a smoothness assumption on the loss function ℓ. We believe that this assumption can be relaxed to a continuity assumption with a little bit more work. But for ease of exposition, we avoid it. In particular, note that we do not assume any bound on the Lipschitz constant of ℓ, which is quite unusual for convergence analysis in the optimization literature.

2. The dependence on all parameters is logarithmic. In contrast, the dependence on the relevant parameters (like Lipschitz or Polyak- Lojasiewicz constant) is at least linear for gradient descent and its stochastic counterparts. Moreover, the dependence on ϵ_0 for stochastic procedures is also always at least linear in $1/\epsilon_0$ even under very limited setting of convex functions [39].

3. The analysis is non-local in the sense that at each iteration we directly track progress with respect to the global minimum value α. In typical analysis in the non-convex optimization literature one uses bounds on the difference between consecutive iterates, i.e, $\ell(x_i) - \ell(x_{i-1})$.

2.4 Main theoretical insight

In this section we state and discuss Lemma 2.2 which forms our main theoretical technique. We state Lemma 2.2 more generally than is needed to prove Theorem 2.1. It is stated for any locally compact Lie group that satisfies Kazhdan's Property (T). We do this in order to emphasize the general nature of our result and to bring out the connection of this crucial lemma with Kazhdan's Property (T) which is a very important and extensively studied property of Lie groups [7]. Note that all locally compact group with a normalized Haar measure are compact. So Lemma 2.2 is equivalent to it's Corollary 2.1 presented in the next subsection. But stating them as two different statements gives us an opportunity to provide two different proofs and highlight the connection of our work with Kazhdan's Property (T). A proof of Lemma 2.2 is presented in Appendix 2.7.1.

Lemma 2.2. *Let G be a locally compact Lie group that satisfies Kazhdan's Property (T) with constant c. Fix a normalized left-invariant Haar-measure on G. Let $f : G \to \mathbb{R}$ be a smooth function such that $\int_G f = 0$. Let $\alpha = \min_{g\in G} f(g)$, $\beta = \max_{g\in G} f(g)$ and $\epsilon = \frac{c\|f\|_2^2}{2|\beta-\alpha|^2}$. Then,*

$$\left|\{g : f(g) - \alpha \leq (1 - \sqrt{\epsilon})(\beta - \alpha)\}\right| \geq \epsilon/2.$$

Contextualizing Lemma 2.2. The lemma gives a non-trivial lower bound on the probability of finding a point that is substantially away from the maximum of the function defined on a locally compact group G, by simply sampling a point randomly according to the fixed left-invariant Haar measure. The fundamental nature of this lemma should be compared with results like the Markov inequality or the Chebyshev inequality, which give a non-trivial lower bound on the probability of getting a value close to the mean by sampling a point according to the used probability distribution.

Generality of Lemma 2.2. Though this result is stated on a Lie group one can transfer it to other spaces which lack this structure, for example, the n-dimensional hypercube. This is possible because one can construct a smooth map from the n-dimensional hypercube to the n-dimensional torus, which is a compact Lie group. We state this here to demonstrate the generality of Lemma 2.2 but we do not provide the details because we do not use such a result in the paper. In the next section, in Lemma 2.3, we discuss how the result can be transferred to an appropriate quotient of a Lie group. We also note that an argument similar to the proof of Lemma 2.2 can be constructed for a discrete group like the boolean hypercube, further increasing the applicability of our result.

2.4.1 Using Lemma 2.2 to prove Theorem 2.1

In Algorithm 2.1 we sample from the Grassmannian, which is a quotient space of the compact Lie group $O(d)$. We do not sample from the group directly. In Lemma 2.3 we show that a statement similar to Lemma 2.2 holds for our quotient space, also. The proof of Lemma 2.3 (which is presented in Appendix 2.7.3) uses Lemma 2.2 adapted to the special case of compact Lie groups (presented in Corollary 2.1). Kazhdan's Property (T) is a concept for Lie groups and does not have an equivalent statement for their quotients.

Corollary 2.1. *Let G be a compact Lie group and let $f : G \to \mathbb{R}$ be a smooth function such that $\int_G f = 0$. Let $\alpha = \min_{g\in G} f(g)$, $\beta = \max_{g\in G} f(g)$ and $\epsilon = \frac{\|f\|_2^2}{|\beta-\alpha|^2}$. Then,*

$$\left|\{g : f(g) - \alpha \leq (1 - \sqrt{\epsilon})(\beta - \alpha)\}\right| \geq \epsilon/2. \tag{2.1}$$

Lemma 2.3. *Let G be a compact Lie group and H a closed subgroup of G. Let $f : G/H \to \mathbb{R}$ be a smooth function such that $\int_{G/H} f = 0$. Let $\alpha = \min_{x\in G/H} f(x)$, $\beta = \max_{x\in G/H} f(x)$ and $\epsilon = \frac{\|f\|_2^2}{|\beta-\alpha|^2}$. Then,*

$$\left|\{x : f(x) - \alpha \leq (1 - \sqrt{\epsilon})(\beta - \alpha)\}\right| \geq \epsilon/2.$$

A direct proof of Corollary 2.1 (which also establishes Lemma 2.1 for compact Lie groups) is presented in Appendix 2.7.2 and the proof for Lemma 2.3 is presented in Appendix 2.7.3.

Discussion on the gap parameter. One of the very important application of Kazhdan's property (T) is the first explicit construction of an expander graph in [80]. By the virtue of their spectral gap (the difference between the first and second eigenvalue of the Laplacian), expander graphs have very good mixing properties, i.e., a random walk on an expander graph quickly gets distributed evenly across the graph [91]. The parameter ϵ in the Lemmas 2.2-2.3 behaves very similarly to the spectral gap of an expander graph. It dictates how fast f can approach its minimum α. In fact, it plays a similar role in the proof of Theorem 2.1 as the spectral gap does in the rapid mixing proofs. More specifically, the key parallel with expander graphs is that their graph adjacency matrix shrinks functions which are orthogonal to constants (e.g., Lemma 1 of Miller and Venkatesan [83]). This is the same operating principle as in Lemmas 2.2-2.3. This is the reason why we call θ the gap function of ℓ.

One can potentially develop this connection with random walks on expander walks further by noticing that our random walk can be modeled using a supermartingale. One can then try to show that as the random walk approaches convergence, it endows a uniform or a near-uniform distribution over the set of all global minima. In an expander random walk this uniform distribution is over the set of all nodes. In a certain sense, because we are always picking a random $G_{k,d}$ (defining the set of hyperplanes containing x_i in the ith iteration of the Algorithm 2.1), our random walk gives a similar sampling strategy, but over the set of all global minima, as does a random walk on an expander graph.

2.5 Robustness

For any smooth function ℓ, by Theorem 2.1 we know that Algorithm 2.1 converges towards its minimum α. However, in practice, the algorithm might converge to a point different

from the global minimum (see Section 2.6). In this section, we discuss why this can happen and what this means for the robustness of the solution obtained from Algorithm 2.1 under perturbations in the training data. The noise model we use in this section was described in Section 2.2.

2.5.1 Ignoring a small set

Let f be a function on the Grassmannian. Consider the situation where there is a set U of small measure on which the function dips dramatically compared to the measure of U. In this case, the minimum of f outside of U may be substantially larger than the minimum of f over its entire domain. The variance of f on this restricted space might still be almost the same as its variance on its entire domain. By only considering the space outside U, the gap parameter increases substantially. This means that the value of f, at a random point on the Grassmannian, will have a higher probability of being close to the minimum outside of U than the one on the entire space. Mathematically, this can be formalized as follows:

Lemma 2.4. *Let G be a compact Lie group with a normalized measure. Let H be a closed subgroup of it. Let G/H, the quotient of G with respect to H, have a normalized measure on it. Let $f : G/H \to \mathbb{R}$ be a smooth function such that $\int_{G/H} f = 0$. Let $\alpha = \min_{x \in G/H} f(x)$, $\beta = \max_{x \in G/H} f(x)$ and $\alpha' \in (\alpha, \beta)$. Set $U = \{x : f(x) < \alpha'\}$ and $\epsilon = \frac{\|f\|_2^2}{|\beta - \alpha'|^2} - 2|U|\frac{|\beta-\alpha|^2}{|\beta-\alpha'|^2}$. Assume $\epsilon > 0$. Then,*

$$\left|\{x : f(x) - \alpha' \leq (1 - \sqrt{\epsilon})(\beta - \alpha')\}\right| \geq \epsilon/2.$$

One way of interpreting this is that random sampling is blind to the bad behavior of the function on small sets in its domain. Leaving out the small set, we get a better gap parameter for the minimizers of our loss function ℓ that lie outside this set. In the next two subsections, we will discuss the implications of this for the robustness of Algorithm 2.1. But first we use Lemma 2.4 to give a new convergence result.

Define a function $\ell_{\alpha'}$ as $\ell_{\alpha'} := \max(\ell, \alpha')$ for some $\alpha' > \alpha$. With Lemma 2.4 in tow, we can now study the convergence properties of Algorithm 2.1 towards α' even when the algorithm uses ℓ in its execution. We therefore obtain the following theorem:

Theorem 2.2. *Let $\ell : \mathbb{R}^d \to \mathbb{R}$ be a smooth loss function and $\alpha = \min_x \ell(x)$. Choose $\alpha' > \alpha$ and set $\ell_{\alpha'} := \max(\ell, \alpha')$. Let $\theta_{\ell_{\alpha'}}$ be the gap function of $\ell_{\alpha'}$. Assume $\theta_{\ell_{\alpha'}} \geq 1 - \delta$ for some $\delta > 0$. Then, for all ϵ_0 and γ in $(0, 1)$, with $N = \frac{\log 1/\epsilon_0}{\log 2/\delta}$ and $T = \frac{\log N + \log 2/3\gamma}{\log 1/\delta}$, with probability at least $1 - \gamma$, Algorithm 2.1 finds an $x \in \mathbb{R}^d$ such that*

$$\ell(x) - \alpha' \leq \epsilon_0(\ell(x_0) - \alpha').$$

Note that $\ell_{\alpha'}$, as defined, might not be a smooth function. But that does not matter since we only use it to compute $\theta_{\ell_{\alpha'}}$ theoretically. It has arbitrarily close smooth approximations that yield the same θ.

2.5.2 Gap parameter as a measure of robustness

In the last section, we saw that leaving a "part" of the function out can increase the gap parameter of the minimizers of the loss function ℓ. In general, the value of ℓ on it's domain

Algorithm 2.2 Our Robust Random Walk

Input: $\ell : \mathbb{R}^d \to \mathbb{R}, x_0 \in \mathbb{R}^d, 1 < k < d, 0 < \theta_0 < 1/2, N > 0$

$T \leftarrow \frac{2N}{\log 1/(1-\theta_0)}$

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

  Sample $\eta_1, \ldots, \eta_T$ uniformly from $G_{k,d}$

  $y_j \leftarrow \arg\min_{y \in x_{i-1}+\eta_j} \ell(y)$ for $j \in [T]$

  $x_i \leftarrow \operatorname{argmin}_{y \in \{y_1, \ldots, y_T\}} \ell(y)$

return x_N

can vary between the maximum and minimum value of ℓ. When set to the maximum value, the gap parameter for the corresponding solutions will be 1 and when set to the minimum value, it will have the smallest possible value for this function. We hypothesize that for a solution x returned by Algorithm 2.1, its gap parameter dictates its robustness as a minimizer of ℓ.

When an adversary introduces a perturbation Δ to the data matrix, if it is able to corrupt the solutions on most of $G_{k,d}$ then the loss function is highly unstable, and there is little hope to build any protection against perturbations. But if we look at the class of loss functions for which most of this perturbation is limited to a small subset of $G_{k,d}$, then for such functions it is natural to aim to find solutions which lie outside of these easily corruptible subsets. Since, by Lemma 2.4, the gap parameter directly measures the size of the set that lies close to a given target value α', if this set is small, it makes the solutions corresponding to this target value more susceptible to noise and hence less robust. This is why it is reasonable to use the gap parameter as a measure of robustness. With this motivation we give a modification of Algorithm 2.1 which can be used to optimize ℓ up to an α' with a desired gap parameter. We present this in Algorithm 2.2 and prove that it finds the correct α' in Theorem 2.3. Note that Algorithm 2.2 does not need α' as an input parameter.

Theorem 2.3. *Let $\ell : \mathbb{R}^d \to \mathbb{R}$ be a smooth loss function. Then for all $N > 0$ and $0 < \theta_0 < 1/2$, Algorithm 2.2, with probability at least $1 - 3/2N$, converges to an α' with $\theta(\ell_{\alpha'}) \geq \theta_0$, i.e., it finds an x such that*

$$\ell(x) - \alpha' \leq \left(1 - \sqrt{2\theta_0}\right)^N \left(\ell(x_0) - \alpha'\right).$$

We provide a proof of this theorem in Appendix 2.7.7.

2.5.3 Dependence of robustness on k

Up until now, we have discussed the convergence properties of the random walk, and identified the gap parameter as an important parameter controlling both the convergence and the robustness of the solution. In this section, we discuss how the choice of k, the dimension of the planes in which the optimization problem is solved, affects the algorithm and in turn informs the gap parameter of the solution retrieved. This subsection is best read in conjunction with Section 2.6 where our experimental results are presented.

A general trend in our experiments, across a range of models, is that for smaller values of k the learned models usually have very good loss values and robustness properties. As k increases, the loss might improve, but at the cost of decreased robustness. For example, in experiments with neural networks, the models learned with a smaller value of k do drastically better on backdoor attacks than the models learned without Algorithm 2.1 while achieving similar accuracy to the latter on clean test data.

As k decreases, the way the optimization problem is adapted to the respective Grassmannian changes, seemingly hiding solutions which are more susceptible to noise in the small sets as discussed in the last two sections. Surprisingly, the solutions retrieved still have close to optimal loss values. We believe that this robust behavior can be attributed to the difficulty of constructing perturbations which can simultaneously affect a large portion of random projections of the data matrix. Choosing k appropriately, we can control the trade-off between obtaining a solution with an optimal loss value and a solution with better robustness properties.

2.6 Experiments

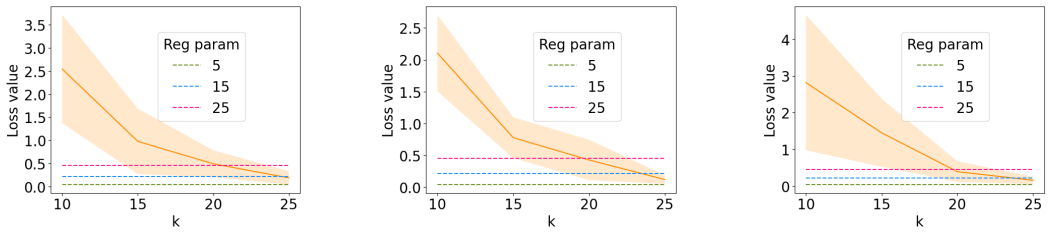
In this section, we show the versatility of our technique by testing it on a wide range of models: Linear Regression, Logistic Regression, SVMs and Neural Networks. We use both synthetic as well as popular evaluation datasets.

Implementation details. To simplify the implementation, we work with a modification of Algorithm 2.1 for our experiments. This modification is presented as Algorithm 2.3 in Appendix 2.7.8. It replaces hyperplanes in Algorithm 2.1 with subspaces, which are hyperplanes that pass through the origin.

Picking a random subspace. One important step in Algorithm 2.3, used in all the experiments below, is that of picking a random subspace containing a given vector $x \in \mathbb{R}^d$. To do this, we consider two different techniques:

1. In the first technique, we start by constructing a basis U for the space orthogonal to x by taking the singular vectors corresponding to non-trivial singular values of the matrix $\mathbb{I}_d - xx^T/\|x\|^2$, where $\mathbb{I}_d$ is the $d \times d$ identity matrix. We then sample a mean 0 and variance 1 gaussian i.i.d. matrix of size $(d-1) \times (d-1)$ and construct $V \in \mathbb{R}^{(d-1)\times(k-1)}$, the matrix whose columns are the top $k-1$ left singular vectors of the randomly sampled matrix. Our desired random supspace then is the span of the column space of UV combined with x.
2. In the second technique, we start by constructing a $d \times k$ matrix U by keeping its first column as x and filling the rest of it's entries with gaussian i.i.d. random variables. We then do a QR decomposition on U and use the orthonormal matrix obtained from this decomposition in our algorithms. Note that the span of this orthonormal matrix will always contain x.

While the first method will provably generate a uniformly random subspace containing x, the second method has no such guarantees. But the second method is computationally much faster when d is large and hence is used for all our deep learning experiments.



(a) Condition number = 10^2 (b) Condition number = 10^4 (c) Condition number = 10^5

Figure 2.1: Plots for Algorithm 2.3 run on Linear Regression. We compare the loss of the solution retrieved for different values of k with the loss of the solutions retrieved by ridge regression with regularization parameters set to 5, 15 or 25. The dark lines correspond to the mean and the shaded area to one standard deviation over 10 runs of the experiment. We see that the linear regression models retrieved by Algorithm 2.3 have losses comparable to those of the regularized models learned with ridge regression.

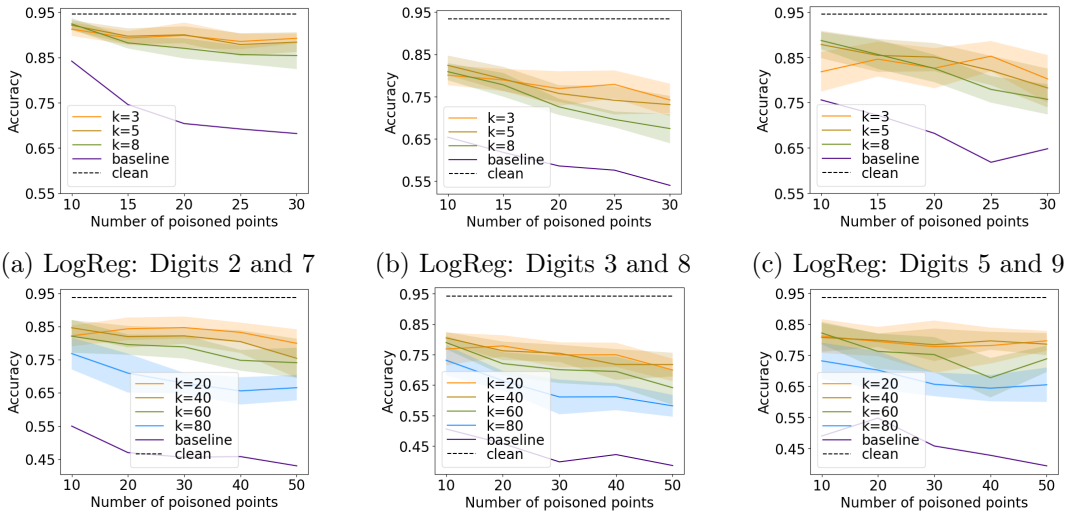
2.6.1 Linear Regression

For linear regression experiments, we work with synthetic data in 100 dimensions with 1000 data points. The behavior of a linear regression instance is largely determined by the condition number of its data matrix. Accordingly, we study the effect of our algorithm for data matrices with preselected condition numbers.

For a given condition number, we generate an instance whose singular values are equally spaced between a top singular value of 100 and the corresponding least singular value. We generate a regressor vector by setting the last five values to 1 and by picking other coordinate uniformly at random between 0 and 1. The idea here is that in real world data, the top singular vectors usually correspond to the signal whereas the last singular vectors correspond to the noise. We might be able to get a solution with a lower loss by fitting to the last singular vectors, but this would be overfitting to the training data. We can avoid this by using some regularization technique like ridge regression (see Section 3.4.1 in Hastie et al. [47]). Using this setting, we want to demonstrate that for an appropriate choice of k, Algorithm 1 retrieves solutions which have loss corresponding to different choices of the regularization parameters in ridge regression. We repeated the experiments 10 times and report the mean and standard deviation in our plots. The results are presented in Figure 2.1. This shows that linear regression models trained with Algorithm 2.3, avoid fitting to the noise in the problem, and hence can be expected to have robust behavior.

2.6.2 Logistic Regression and SVMs

For binary classification experiments, we use a subset of the MNIST dataset by sampling 100 images corresponding to a pair of digits to construct our training dataset, and 500 images to construct our testing dataset. We then use SecML [82], a library for secure and explainable Machine Learning in Python, to poison the training dataset to degrade the performance of the learned classifier. The library implements the attack from [27] to generate poisoned datasets for logistic regression and the attack from [13] for SVMs. We study the effect of poisoning an increasing number of points on various choices of k for Algorithm 2.3. As a baseline, we compare this with the accuracy obtained by training the



(a) LogReg: Digits 2 and 7 (b) LogReg: Digits 3 and 8 (c) LogReg: Digits 5 and 9

(d) SVM: Digits 2 and 7 (e) SVM: Digits 3 and 8 (f) SVM: Digits 5 and 9

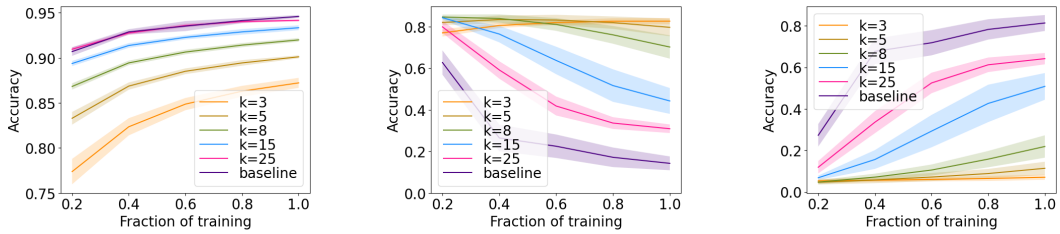
Figure 2.2: Plots for classifying pairs of digits from MNIST dataset using the logistic regression and SVM models trained with Algorithm 2.3. We poison the datasets using SecML [82] and compare the accuracy of a solution retrieved by Algorithm 2.3, for various values of k, to the solution obtained by directly learning the classifier on the poisoned dataset (this corresponds to the baseline). For reference, we also give the accuracy of the model trained on the clean data in the plots. The dark lines correspond to the mean and the shaded area to one standard deviation over 10 runs of the experiment. We see across all the plots that training with Algorithm 2.3 yields models with substantially better accuracy in presence of the data poisoning attacks.

corresponding classifiers without Algorithm 2.3. We also give the accuracy for training the classifiers without Algorithm 2.3 on a dataset with no poisoned samples. We repeated the experiments 10 times and report the mean and standard deviation in our plots. The results are presented in Figure 2.2. As we can see, the models obtained from the training with Algorithm 2.3 give much better accuracy than those trained without it. Observation also indicates that the accuracy is generally better for smaller values of k. We note that SVM is not a "smooth" optimization problem per se, but Algorithms 2.1, 2.2 and 2.3 are still well defined for it.

2.6.3 Neural Networks

In this section, we discuss the efficacy of Algorithm 2.3 against backdoor attacks in deep learning. The agenda of a backdoor attack is to emanate a specific response from a trained network when a test image has a special patch of pixels (the backdoor) overlapped on it. This attack can be used to misclassify images during testing. To carry out such an attack, an adversary introduces a set of images with the backdoor attached to them, and with their labels set to a desired label in the training dataset. The network then runs the risk of learning an association between the backdoor and the desired label, while ignoring the true label of the image entirely.

Attack from [100]. For our experiments, we use the implementation of this attack pro-



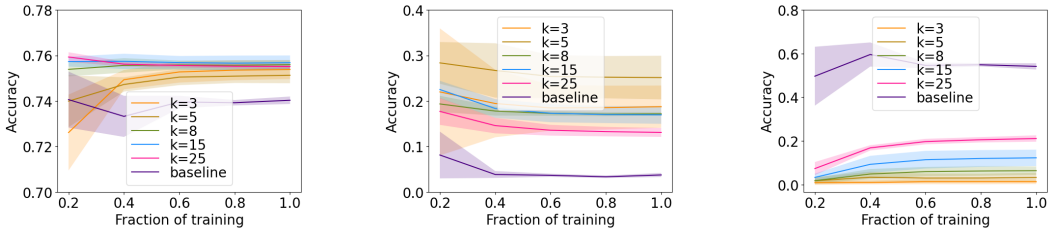
(a) Accuracy on a clean test set (b) Accuracy on a poisoned test set (c) Accuracy of the poisoning attack

Figure 2.3: Accuracy plots for MNIST against a backdoor attack presented in [100]. A feedforward neural network is trained with Algorithm 2.3 for different values of k with poisoned samples in the training data. We report three metrics: 1) accuracy on a clean test set which doesn't contain any images with the backdoor, 2) accuracy on a poisoned test set which contains images with the backdoor, 3) accuracy of the attack, i.e., images with the backdoor getting classified as intended by the adversary. We compare the results against the same model trained directly (this corresponds to the baseline). The dark lines correspond to the mean and the shaded area to one standard deviation over 5 runs of the experiment. At 0.3 fraction of the training, a modest decrease in the clean accuracy of the models trained using Algorithm 2.3 yields substantially better accuracy on the poisoned data set while also considerably decreasing the accuracy of the attack.

vided in the ART toolbox [87]. In this attack the backdoor is inserted only into images corresponding to a target label. This is done to avoid the filtering of clearly mislabeled poisoned samples by human inspection. We work with the MNIST dataset. Our model is a fully connected MLP with three hidden layers and 100 neurons in each layer. For training, we use the Adam optimizer. A baseline model is trained on the poisoned dataset for 10 epochs. For training with Algorithm 2.3, we set $N = 10$. To solve the problem in the subspace selected in a given iteration of Algorithm 2.3, we train the network for 5 epochs. The models are evaluated on a clean test set as well as a poisoned test set which consists of images corrupted with the backdoor. We repeated the experiments 5 times and report the mean and standard deviation in our plots. The results are presented in Figure 2.3. The baseline model corresponds to $k = 784$, which is the full dimension of the problem.

Since the parameters of an MLP are distributed across different layers and neurons, treating them as part of the same Euclidean space and working with the subspaces of this single Euclidean space is unnatural. Instead, we work with the parameters of each neuron separately by treating them as living in their own Euclidean spaces, and sampling different subspaces for each of these spaces individually when running Algorithm 2.3. This corresponds to working with a finite product of Grassmannians, which is still the quotient of a compact lie group (the group now will be a product of the same number of orthonormal groups). All theoretical guarantees hold in this setting, since the underlying mathematics is based on compact lie groups.

As we can see in Figure 2.3, the models trained using Algorithm 2.3 are able to achieve an accuracy on the clean test data which is close to that of the accuracy of the models trained without it. At the same time, their accuracy on the poisoned test data is substantially higher and thus the success rate of the poisoning attack substantially smaller. Specifically,



(a) Accuracy on a clean test set (b) Accuracy on a poisoned test set (c) Accuracy on a poisoning attack

Figure 2.4: Accuracy plots for CIFAR-10 against the backdoor attack presented in [93]. A CNN based architecture is fine-tuned with Algorithm 2.3 for different values of k on a training set which contains poisoned data points. We report three metrics: 1) accuracy on a clean test set which doesn't contain any images with the backdoor, 2) accuracy on a poisoned test set which contains images with the backdoor, 3) accuracy of the attack, i.e., images with the backdoor getting classified as intended by the adversary. We compare the results against the same model fine-tuned directly (this corresponds to the baseline). The dark lines correspond to the mean and the shaded area to one standard deviation over 5 runs of the experiment. We see that the models trained with Algorithm 2.3 not only have better accuracy on the clean test set, but also have better accuracy on the poisoned test set and are able to substantially decrease the accuracy of the attack.

around the 1/3rd training mark, the accuracy of the models trained with Algorithm 2.3 with $k \leq 15$ is greater than 80%, while those trained without it have an accuracy of less than 40% on the posioned test data.

Attack from [93]. For this attack too, we use the implementation provided by [87]. The intent of the attack is the same as the previous one. It is constructed in a manner so that the poisoned image is closer to the desired target image in the feature space while visually being indistinguishable from its source image. In [93], the authors show that the attack is robust against many existing defense mechanisms.

For our experiments, we work with the CIFAR-10 dataset and the CNNs-based architecture used by [87] in their demonstration of the attack. We do not attempt to optimize any hyperparameters to improve the clean classification accuracy of the used model. Instead, we choose to work with the experimental setup of [87] to demonstrate the versatility of our technique. In their setup, the poisoned dataset is used only in the fine-tuning step where all but the last fully connected layer (which has a hidden dimension of 4096) are frozen. We use Algorithm 2.3 on this last layer, modifying it in a way similar to what we did in the last section.

We pretrained a model for 200 epochs using SGD with learning rate 0.01, momentum 0.9 and weight decay 2×10^{-4}, reducing the learning rate by a factor of 0.1 after 100 and 150 epochs. For fine-tuning we reinitialize the last layer with gaussian i.i.d. random variables and train for another 10 epochs. For fine-tuning with Algorithm 2.3, apart from reinitializing the last layer, we use $N = 10$ and to solve the problem in the selected subspace of each iteration, we train for 1 epoch. We repeated the experiments 5 times and report the mean and standard deviation in our plots.

We present the results of our experiments in Figure 2.4 and consider three metrics: accuracy on a benign unpoisoned test set, accuracy on a poisoned test set, and the success rate of the attack on this poisoned test set. As we can see Algorithm 2.3 does not affect the accuracy of trained model on the benign samples, while drastically increasing its accuracy on poisoned samples and drastically decreasing the efficacy of the attack on the same samples, especially for smaller values of k.

2.7 Proofs

We present the details left out from the main body of the paper here.

2.7.1 Proof of Lemma 2.2

Proof. Since f is a non-constant smooth function with zero mean, we have, by Kazhdan's Property (T) that there exists a $\gamma \in G$ such that

$$\|f - \gamma \cdot f\|^2 \geq c \cdot \|f\|^2$$

where c is the Kazhdan constant of G. Define $h : G \to [0,1]$ as $h(g) = \frac{|f(g)-\gamma\cdot f(g)|^2}{|\beta-\alpha|^2}$. Let $U_\epsilon \coloneqq \{g : h(g) > \epsilon\}$. Then, we have by Lebesgue integration,

$$\int_0^1 |U_\epsilon| d\epsilon = \int_G h(g) dg = \frac{\|f - \gamma \cdot f\|^2}{|\beta - \alpha|^2} \geq \frac{c \cdot \|f\|^2}{|\beta - \alpha|^2}. \tag{2.2}$$

Since f is a smooth function, $|U_\epsilon|$ is a continuous non-increasing function of ϵ. Moreover, $|U_0| = 1$ and $|U_1| = 0$. From this we have the following upper bound for any $\epsilon' \in [0,1]$,

$$\int_0^1 |U_\epsilon| d\epsilon \leq \epsilon' + |U_{\epsilon'}|. \tag{2.3}$$

Let $\epsilon' \in [0,1]$ be such that $|U_{\epsilon'}| = \epsilon'$. Substituting this ϵ' in (2.3) and using the lower bound from (2.2) we get $\epsilon' \geq \frac{c\cdot\|f\|^2}{2|\beta-\alpha|^2}$. For this ϵ', since $|U_{\epsilon'}| = \epsilon'$, we also have $|U_{\epsilon'}| \geq \frac{c\cdot\|f\|^2}{2|\beta-\alpha|^2}$. As $|U_{\epsilon'}|$ decreases as ϵ' increases, we can select $\epsilon' = \frac{c\cdot\|f\|^2}{2|\beta-\alpha|^2}$ and for this ϵ' we will still have $|U_{\epsilon'}| \geq \frac{c\cdot\|f\|^2}{2|\beta-\alpha|^2}$.

Now, using the definition of h, for every $g \in U_\epsilon$ we have

$$\frac{|f(g) - \gamma \cdot f(g)|^2}{|\beta - \alpha|^2} \geq \epsilon'.$$

On taking the denominator to the right and taking a square root on both the sides, we get either

$$f(g) - \gamma \cdot f(g) > \sqrt{\epsilon'}(\beta - \alpha) \quad \text{or} \quad \gamma \cdot f(g) - f(g) > \sqrt{\epsilon'}(\beta - \alpha).$$

Since both $f(g)$ and $\gamma \cdot f(g)$ are less than β, we can substitute $f(g)$ with it in the first equation and $\gamma \cdot f(g)$ with it in the second equation. We get

$$\beta - \gamma \cdot f(g) > \sqrt{\epsilon'}(\beta - \alpha) \quad \text{or} \quad \beta - f(g) > \sqrt{\epsilon'}(\beta - \alpha).$$

On rearranging, we get

$$\gamma \cdot f(g) < \beta - \sqrt{\epsilon'}(\beta - \alpha) \quad \text{or} \quad f(g) < \beta - \sqrt{\epsilon'}(\beta - \alpha).$$

Now subtract α on both sides to get

$$\gamma \cdot f(g) - \alpha < (1 - \sqrt{\epsilon'})(\beta - \alpha) \quad \text{or} \quad f(g) - \alpha < (1 - \sqrt{\epsilon'})(\beta - \alpha).$$

Since the above statements are strict inequalities, when either of them is true, there will exist a small ball around g contained in U_ϵ such that the corresponding inequality will also be true for every element in this small ball. Let U_ϵ^1 be the union of such balls corresponding to the set for which the first inequality is true, and let U_ϵ^2 be the corresponding union for which the second inequality is true. By construction, both sets are open. Also, they are measurable as the Haar measure is a Borel measure by definition.

Now, every element of U_ϵ will belong to one of these two sets, hence $|U_\epsilon^1| + |U_\epsilon^2| \geq |U_\epsilon|$ and at least one of them will have a measure greater than or equal to $|U_\epsilon|/2$. Since the measure is left-invariant $|\gamma \cdot U_\epsilon^2| = |U_\epsilon^2|$. This gives us the conclusion. □

2.7.2 Proof of Corollary 2.1

Proof. Consider the following integral for a non-constant zero-mean smooth function $f : G \to \mathbb{R}$,

$$\begin{aligned}
\int_G \|f - \gamma \cdot f\|^2 d\gamma &= \int_G \int_G (f(g) - \gamma \cdot f(g))^2 dg d\gamma \\
&= \int_G \int_G \left(f(g)^2 + f(\gamma^{-1} \cdot g)^2 - 2f(g)f(\gamma \cdot g)\right) dg d\gamma \\
&\overset{(a)}{=} \int_G f(g)^2 dg + \int_G \int_G f(\gamma^{-1} \cdot g)^2 dg d\gamma - 2 \int_G f(g) \int_G f(\gamma^{-1} \cdot g) d\gamma dg \\
&\overset{(b)}{=} \int_G f(g)^2 dg + \int_G f(g)^2 dg - 2 \left(\int_G f(\gamma^{-1}) d\gamma\right) \left(\int_G f(g) dg\right) \\
&\overset{(c)}{=} 2 \int_G f(g)^2 dg \\
&= 2\|f\|^2
\end{aligned}$$

where we change the order of integration for the last term in (a), use the invariance property of the Haar measure for compact Lie groups to simplify the second and third term in (b) (left invariance for the second term and right invariance for the third term) and use the fact that f is mean-zero for (c).

Using mean value theorem and the above calculation we see that there exists a $\gamma \in G$ such that,

$$\|f - \gamma \cdot f\|^2 \geq 2\|f\|^2. \tag{2.4}$$

From here on we proceed exactly as we did for the proof of Lemma 2.2 but with fewer details. Define $h : G \to [0, 1]$ as $h(g) = \frac{|f(g) - \gamma \cdot f(g)|^2}{|\beta - \alpha|^2}$. Let $U_\epsilon \coloneqq \{g : h(g) \geq \epsilon\}$. Then, we have by Lebesgue integration,

$$\int_0^1 |U_\epsilon| d\epsilon = \int_G h(g) dg = \frac{\|f - \gamma \cdot f\|^2}{|\beta - \alpha|^2} \geq \frac{2\|f\|^2}{|\beta - \alpha|^2}.$$

Since, $|U_\epsilon|$ is a non-increasing function of ϵ and, $|U_0| = 1$ and $|U_1| = 0$, we have $|U_\epsilon| \geq \frac{\|f\|^2}{|\beta-\alpha|^2}$ for $\epsilon = \frac{\|f\|^2}{|\beta-\alpha|^2}$.

Now, for $g \in U_\epsilon$ we have, either

$$f(g) - \alpha \leq (1 - \sqrt{\epsilon})(\beta - \alpha) \quad \text{or,} \quad \gamma \cdot f(g) - \alpha \leq (1 - \sqrt{\epsilon})(\beta - \alpha).$$

This gives us the conclusion. □

2.7.3 Proof of Lemma 2.3

We use the following lemma on the existence of an invariant measure on quotient spaces for our proof:

Lemma 2.5 (Theorem 1.9 and Remark on page 93 of [49])**.** *Let G be a compact Lie group and H a compact Lie subgroup of G. Then there exists a unique normalized left G-invariant measure dx on G/H such that for all $f \in C(G)$*

$$\int_G f(g)dg = \int_{G/H} \int_H f(x \cdot h)dhdx$$

where dg and dh are normalized left-invariant measures on G and H respectively.

Proof of Lemma 2.3. Consider any function $t : G/H \to \mathbb{R}$. Since $G = \bigcup_{h \in H} ((G/H) \cdot h)$, we can define a function $t' : G \to \mathbb{R}$ as $t'(x \cdot h) = t(x), \forall h \in H$ and $x \in G/H$. Define f' using f similarly. Corollary 2.1 gives us an estimate on the measure of the "good" subset of G for f'. We will use U_G to denote this set and $U_{G/H}$ to denote a corresponding set on G/H for f i.e.,

$$U_{G/H} \coloneqq \left\{ x : \frac{|f(x) - \gamma \cdot f(x)|^2}{|\beta - \alpha|^2} \geq \epsilon \right\}$$

where ϵ here is the same as in Corollary 2.1 and, α and β are the minimum and maximum values of f respectively. Note that they are also the minimum and maximum values of f' respectively. This follows from the definition of f'.

Now, let dg, dh and dx be normalized measures on G, H and G/H respectively. Using Lemma 2.5 we can write

$$\begin{aligned}
\int_G t'(g)dg &= \int_{G/H} \int_H t'(x \cdot h)dhdx \\
&= \int_{G/H} \int_H t(x)dhdx \\
&= \left(\int_{G/H} t(x)dx \right) \left(\int_H dh \right) \\
&= \left(\int_{G/H} t(x)dx \right).
\end{aligned}$$

Note that f' is constant on the cosets gH of H in G. This means that for any g in the good set of f', the good set will contain the entire coset gH. Set $t = 1_{\{x \in U_{G/H}\}}$ in the above calculation, then $t' = 1_{\{g \in U_G\}}$. We get that measure of the good set of f on G/H will be the same as the measure of the good set of f' on G. This gives us the conclusion. □

2.7.4 Proof of Lemma 2.4

Proof of Lemma 2.4. We first prove an equivalent statement over the group G in Lemma 2.6, then we can use the same machinery as we did in proving Lemma 2.3 from Corollary 2.1 to transfer the estimate from the group to it's quotient to get the full proof. The details are straightforward. □

Lemma 2.6. *Let G be a compact Lie group and let $f : G \to \mathbb{R}$ be a smooth function such that $\int_G f = 0$. Let $\alpha = \min_{g\in G} f(g)$, $\beta = \max_{g\in G} f(g)$ and $\alpha' \in (\alpha, \beta)$. Set $U = \{g : f(g) < \alpha'\}$ and $\epsilon = \frac{\|f\|_2^2}{|\beta-\alpha'|^2} - 2|U|\frac{|\beta-\alpha|^2}{|\beta-\alpha'|^2}$. Then,*

$$\left|\{g : f(g) - \alpha' \le (1 - \sqrt{\epsilon})(\beta - \alpha')\}\right| \ge \epsilon/2$$

Proof. The proof proceeds in a manner similar to that of Corollary 2.1. We provide the extra details needed here. Define $h : G \to [0, 1]$ as $h(g) = \frac{|f(g)-\gamma\cdot f(g)|^2}{|\beta-\alpha'|^2}$ and let $V = U \cup \gamma^{-1} \cdot U$. We consider integrals over the space $G \setminus V$. To do this we use the normalized measure dg on G and divide it by $|V|$ so that the resulting measure is normalized over $G \setminus V$. We denote this measure by dg_V. We use $|\cdot|_V$ to denote the size of a set w.r.t. this measure.

Now, let $U_\epsilon \coloneqq \{g : h(g) \ge \epsilon, g \in G \setminus V\}$. We use the measure dg_V when measure the size of the set U_ϵ. We have,

$$\begin{aligned}
\int_0^1 |U_\epsilon|_V d\epsilon &\overset{(a)}{=} \int_{G\setminus V} h(g) dg_V \\
&= \int_{G\setminus V} \frac{|f(g) - \gamma \cdot f(g)|^2}{|\beta - \alpha'|^2} dg_V \\
&\overset{(b)}{=} \frac{1}{|\beta - \alpha'|^2} \left(\int_{G\setminus V} \frac{|f(g) - \gamma \cdot f(g)|^2}{|G \setminus V|} dg \right) \\
&= \frac{1}{|G \setminus V||\beta - \alpha'|^2} \left(\int_G |f(g) - \gamma \cdot f(g)|^2 dg - \int_V |f(g) - \gamma \cdot f(g)|^2 dg \right) \\
&\overset{(c)}{\ge} \frac{\|f - \gamma \cdot f\|^2 - |V||\beta - \alpha|^2}{|G \setminus V||\beta - \alpha'|^2} \\
&\overset{(d)}{\ge} \frac{2\|f\|^2 - |V||\beta - \alpha|^2}{|G \setminus V||\beta - \alpha'|^2}
\end{aligned}$$

where we use Lebesgue integration in (a), we change the measure from dg_V to dg in (b), use the upper and lower bound on f to get (c) and use (2.4) to get (d).

Since, $|U_\epsilon|_V$ is a non-increasing function of ϵ and, $|U_0|_V = 1$ and $|U_1|_V = 0$, using the same ideas as in proof of Corollary 2.1 we have $|U_\epsilon|_V \ge \frac{2\|f\|^2-|V||\beta-\alpha|^2}{2|G\setminus V||\beta-\alpha'|^2}$ for $\epsilon = \frac{2\|f\|^2-|V||\beta-\alpha|^2}{2|G\setminus V||\beta-\alpha'|^2}$. Moreover, since $|U_\epsilon|_V = \frac{|U_\epsilon|}{|G\setminus V|}$, we have $|U_\epsilon| \ge \frac{\|f\|^2}{|\beta-\alpha'|^2} - |V|\frac{|\beta-\alpha|^2}{|\beta-\alpha'|^2} \ge \frac{\|f\|^2}{|\beta-\alpha'|^2} - 2|U|\frac{|\beta-\alpha|^2}{|\beta-\alpha'|^2}$.

Now, for $g \in U_\epsilon$ we have, either

$$f(g) - \alpha' \le (1 - \sqrt{\epsilon})(\beta - \alpha') \quad \text{or,} \quad \gamma \cdot f(g) - \alpha' \le (1 - \sqrt{\epsilon})(\beta - \alpha').$$

Using the same argument as in the proof of Corollary 2.1 we get the conclusion. □

2.7.5 Proof of Theorem 2.1

Proof. We use the notation set up in Section 2.3.2 for this proof. At step i of Algorithm 2.1, from Lemma 2.3, we know that we can find an η_i such that

$$L(x_{i-1}, \eta_i) - m(x_{i-1}) \leq \left(1 - \sqrt{2\Theta(x_{i-1})}\right) (M(x_{i-1}) - m(x_{i-1})) \tag{2.5}$$

with probability at least $\Theta(x_{i-1})$. Since $\Theta(x_{i-1}) \geq \theta(\ell) \geq 1 - \delta$ and, as we sample T points in each iteration and take the minimum over these samples, the probability that we will find one such point amplifies to $1 - \delta^T$.

The probability of this happening for all N iterations of the algorithm is $(1 - \delta^T)^N$. We want this probability to be greater than $1-\gamma$. Set $(1-\delta^T)^N \geq 1-\gamma$ and take the logarithm on both sides. Rearrange, and we get $N \log \frac{1}{1-\delta^T} \leq \log \frac{1}{1-\gamma}$. Now, we use the following approximations to simplify further:

$$\forall t \in [0, 1), \quad t \leq \log \frac{1}{1-t} \leq t + \frac{t^2}{2} \leq \frac{3t}{2}.$$

Using these approximations it is sufficient to work with $N\delta^T \leq 3\gamma/2$. Taking logarithm on both the sides again and rearranging we get,

$$T \geq \frac{\log N + \log 2/3\gamma}{\log 1/\delta}.$$

This gives us a bound on the number of samples we need to draw in each iteration of Algorithm 2.1.

Now, we need two facts to proceed:

1. $m(x)$ is a constant function with value α
2. $\forall i \in [1, T],\ M(x_i) \leq L(x_{i-1}, \eta_i)$.

To prove the first we proceed as follows. Recall $\alpha = \min_x \ell(x)$. Then for $k \geq 2$, $\forall x,\ m(x) = \alpha$. This is because, for any given x, there exists a k-plane that passes through x and a global minimum of ℓ.

To prove the second, notice that x_i is an argmin of ℓ in the k-plane $x_{i-1} + \eta_i$. This k-plane will correspond to some $\eta \in G_{k,d}$ such that $x_i + \eta \cong x_{i-1} + \eta_i$. Moreover, on any other k-plane that contains x_i the minimum value of ℓ will be upper bounded by $\ell(x_i)$. Hence $M(x_i) = \max_\eta L(x_i, \eta) \leq \ell(x_i) = L(x_{i-1}, \eta_i)$.

Using the above two facts we can rewrite (2.5) as,

$$\ell(x_i) - \alpha \leq \left(1 - \sqrt{2\Theta(x_{i-1})}\right) (\ell(x_{i-1}) - \alpha).$$

Conjugating this over all N steps we get,

$$\ell(x_N) - \alpha \leq \prod_{i=1}^{N} \left(1 - \sqrt{2\Theta(x_i)}\right) (\ell(x_0) - \alpha).$$

For convenience, we loosen this equation a bit by dropping the 2 in the equation and substituting $\Theta(x)$ with $1-\delta$ to get,

$$\ell(x_N) - \alpha \leq \left(1 - \sqrt{1-\delta}\right)^N (\ell(x_0) - \alpha).$$

We want $\left(1 - \sqrt{1-\delta}\right)^N \leq \epsilon_0$. This gives us $N \geq \frac{\log \epsilon_0}{\log\left(1-\sqrt{1-\delta}\right)}$. This can be further simplified as follows:

$$\begin{aligned} N &\geq \frac{\log \epsilon_0}{\log(1 - \sqrt{1-\delta})} \\ &\overset{(a)}{\geq} \frac{\log \epsilon_0}{\log \delta/2} = \frac{\log 1/\epsilon_0}{\log 2/\delta} \end{aligned}$$

where we use the fact that $\sqrt{1-\delta} \leq 1 - \delta/2$ for $\delta \geq 0$ in (a). This completes the proof. □

2.7.6 Proof of Theorem 2.2

Proof. The proof here is exactly the same as the proof of Theorem 2.1. □

2.7.7 Proof of Theorem 2.3

Proof. Let $\beta = \max_x \ell(x)$, then $\theta(\ell_\beta) = 1$. Now, if $\theta(\ell_\alpha) \geq \theta_0$, the theorem is trivially true. So we suppose that this is not the case. Since θ as a function of α' is continuous there exists an α' such that $\theta(\ell'_\alpha) \geq \theta_0$.

Now, let $\theta_0 = 1 - \delta$. In step i of Algorithm 2.2, with probability greater than $1 - \delta^T$ we find an x such that

$$\ell(x_i) - \alpha' \leq \left(1 - \sqrt{2(1-\delta)}\right) (\ell(x_{i+1}) - \alpha').$$

By composition, after N with probability greater than $(1 - \delta^T)^N$ we have,

$$\begin{aligned} \ell(x_N) - \alpha' &\leq \left(1 - \sqrt{2(1-\delta)}\right)^N (\ell(x_0) - \alpha') \\ &= \left(1 - \sqrt{2\theta_0}\right)^N (\ell(x_0) - \alpha'). \end{aligned}$$

Now, we need to lower bound the probability of success $(1 - \delta^T)^N$. To do so we consider the negative logarithm of this quantity,

$$\begin{aligned} -N \log\left(1 - (1-\theta_0)^T\right) &= N \log\left(\frac{1}{1 - (1-\theta_0)^T}\right) \\ &\overset{(a)}{\leq} \frac{3N(1-\theta_0)^T}{2} \\ &= \frac{3N}{2} 2^{T \log(1-\theta_0)} \end{aligned}$$

Algorithm 2.3 Random Walk for the experiments

Input: $\ell : \mathbb{R}^d \to \mathbb{R}, x_0 \in \mathbb{R}^d, 1 < k < d, N > 0$

1: **for** $i = 1, \ldots, N$ **do**
2: $\quad \bar{x}_{i-1} \leftarrow \frac{x_{i-1}}{\|x_{i-1}\|}$
3: $\quad$ Sample η uniformly from $G_{k-1,d-1}$
4: $\quad x_i \leftarrow \arg\min_{y \in \pi(\bar{x}_{i-1}, \eta)} \ell(y)$
5: **return** x_N

where (a) follows from the inequality $\log \frac{1}{1-t} \leq \frac{3t}{2}$ for all $t > 0$. Setting $T = \frac{2 \log N}{\log 1/(1-\theta_0)}$, we get

$$\begin{aligned} -N \log \left(1 - (1-\theta_0)^T\right) &\leq \frac{3N}{2} 2^{-2 \log N} \\ &\leq \frac{3N}{2} \frac{1}{N^2} = \frac{3}{2N}. \end{aligned}$$

Hence, the probability of success is at least $2^{-3/2N} \geq 1 - 3/2N$ for all $N > 0$. This gives us the theorem. □

2.7.8 Implementation details

Instead of working with Algorithms 2.1 and 2.2 as they are, we modify them a bit to make them more implementation friendly. To do this, we make two modifications:

1. First, redefine the function L defined in Section 2.3.2 by changing its domain. To do this, define $\pi : G_{1,d} \times G_{k-1,d-1} \to G_{k,d}$ as described now. For (x, η) in the domain of π pick a fixed basis, represented by a matrix $U \in \mathbb{R}^{d \times (d-1)}$, for the space $x^\perp$ orthogonal to x in $\mathbb{R}^d$. Note that $x^\perp$ is a $(d-1)$-dimensional space. Pick η from $G_{k-1,d-1}$ and use a matrix $V \in \mathbb{R}^{(d-1) \times (k-1)}$ to represent it as a subspace of $\mathbb{R}^{d-1}$. Then construct a $(k-1)$-dimensional subspace of $x^\perp$ by considering the subspace spanned by UV. Note that this subspace will live in $\mathbb{R}^d$ and will be orthogonal to x. The image of (x, η) under π is the k-dimensional space spanned by x and $\eta_{x^\perp}$.

   Now, define $L : G_{1,d} \times G_{k-1,d-1} \to \mathbb{R}$ as follows:

$$L(x, \eta) \coloneqq \min_{y \in \pi(x,\eta)} \ell(y)$$

2. Second, we do not sample multiple subspaces in each iteration. This decreases the computational complexity of the algorithm and is motivated by empirical observations.

We present the modification of Algorithm 2.1 that we use in our experiments in Algorithm 2.3.

The reason why we do not work with this formulation is to avoid using too many unnecessary theoretical concepts that might obscure intuition. To theoretically analyze Algorithm 2.3, mathematically the correct manifold to use is a degenerate flag manifold [69] instead of

$G_{1,d} \times G_{k-1,d-1}$. The theoretical analysis still remains the same as it is mostly concerned with the use of the Grassmannian as the second space in the product manifold. However, this version of the algorithm is much easier to implement since it eliminates the affine component present in Algorithms 2.1 and 2.2.

Part II

New perspectives on Euclidean optimization

3 | Euclidean optimization on the Grassmannian

Let $f : \mathbb{R}^d \to \mathbb{R}$ be the function that we want to minimize. Let us assume that we are given a black box that solves any Euclidean minimization problem of dimension $k < d$. How can we use this black box to solve the d-dimensional minimization problem defined by f? We studied one approach to this problem in Chapter 2, but as we saw, that approach has an inherent robustness because of which it might not always yield the true optimum of f. In this chapter, we develop techniques which differ from those in Chapter 2 in the sense that they aim to find a true minimum of f.

We proceed in a way similar to Section 2.3.2 by defining an alternative optimization problem. But instead of working with $\mathbb{R}^d \times G_{k,d}$ we work only with $G_{k,d}$. Let us define $F : G_{k,d} \to \mathbb{R}$ as:

$$F(\eta) \coloneqq \min_{x \in \eta} f(x) \tag{3.1}$$

As it turns out, $G_{k,d}$ has the structure of a differentiable manifold which can be used to construct an array of familiar optimization techniques on it. For our purpose of optimizing F we will work with gradient descent as outlined in Algorithm 1.1.

In this chapter, we first discuss the differential geometry of the Grassmannain in Section 3.1, then we present the various calculations needed to realize the gradient descent procedure with the specified geometry in Section 3.2. Finally, we present our convergence results in Section 3.3.

The main purpose of this chapter is to demonstrate an alternative perspective on how Euclidean optimization can be approached. We do not compare the technique developed in this chapter with existing techniques in terms of either experimental evaluation or convergence rates.

3.1 Differential geometry of the Grassmannian

We introduced the Grassmannian as the set of all subspaces in Section 2.2 where we studied it as a quotient of the orthogonal group and used the Kazhdan property (T) satisfied by the latter to obtain the main theoretical result of Chapter 2 on the Grassmannian. In this chapter, we will be dealing with the differential geometry of this object.

There are many different perspectives from which the differential geometry of the Grassmannian can be derived [8]. For our purpose, we look at the Grassmannian as a quotient of the non-compact Stiefel manifold. This perspective was developed in [2] and we borrowed most of the material in this section from there.

The noncompact Stiefel manifold $ST_{k,d}$ is the set of all $d \times k$ matrices of full rank. It can be defined as:

$$ST_{k,d} \coloneqq \{Y \in \mathbb{R}^{d\times k} : \text{rank}(Y) = k\}$$

The span of the columns of every $d \times k$ full-rank matrix represents a subspace. Moreover, for a given such matrix Y if we multiply it on the right by any full-rank $k \times k$ matrix, then the subspace defined by the columns of the resultant matrix is the same as that defined by the columns of Y. The collection of all $k \times k$ full-rank matrices forms the General Linear group GL_k which acts over $\mathbb{R}^k$. Hence, by taking the quotient of $ST_{k,d}$ with respect to GL_k we get the Grassmannian $G_{k,d}$, i.e. $G_{k,d} \cong ST_{k,d}/GL_k$. Let $\pi : ST_{k,d} \to G_{k,d}$ be the map that takes a full-rank $d \times k$ matrix to the subspace defined by the span of its columns. Then the quadruple $(GL_k, ST_{k,d}, \pi, G_{k,d})$ is a principal GL_k fiber bundle with total space $ST_{k,d}$ and base space $G_{k,d}$.

The benefit of using this perspective lies in the fact that locally, in an open neighborhood, the Stiefel manifold $ST_{k,d}$ behaves like a Euclidean space $\mathbb{R}^{d\times k}$. That is, it has a trivial embedding in it. For any element $W \in ST_{k,d}$ consider the open ball

$$U = \{Y \in \mathbb{R}^{d\times k} : \|W - Y\|_2 < \lambda_{min}(W)\}$$

where $\lambda_{min}(W)$ is the smallest singular value of W. All elements in U are full rank and therefore belong to $ST_{k,d}$. Thus, around any point $W \in ST_{k,d}$ we can find an open set $U \subseteq ST_{k,d}$ and a map $\phi : U \to \mathbb{R}^{d\times k}$ s.t. ϕ is an identity. This gives us a Euclidean coordinate system around W, and enables $ST_{k,d}$ to inherit the standard metric from the Euclidean space locally. One then obtains an easy-to-work Riemannian geometry for the manifold and derivatives of functions defined on the Stiefel manifold can be computed as their Euclidean derivatives locally. Now we can utilize this Riemannian geometry of $ST_{k,d}$ and the principal GL_k fiber bundle mentioned previously to endow a Riemannian geometry over $G_{k,d}$.

The process of endowing this geometry is certainly nontrivial and has been previously studied in the literature in detail. To gain access to all details of this process, one will have to study the structure of the principal fiber bundles [64, 65], specifically concepts such as cross sections, decomposition of the tangent space of the total space into horizontal and vertical space and horizontal lift of tangent vectors from the tangent space of the base space to that of the total space. One will then have to instantiate these concepts for the specific principal fiber bundle in use by identifying the correct matrix representations of the concerned geometric objects, for example as done in [35] for our case. And then bring everything together as done in [2] to extract the differential geometry for the Grassmannian out of that of the Stiefel manifold via the principal fiber bundle connection.

These concepts are required to build the necessary bridges to enable calculations using a *chosen* full-rank matrix Y as a representation of a subspace, which is an abstract concept, while remaining faithful to the geometry of the abstract manifold $G_{k,d}$. Effectively, we

obtain formulae for various Riemannian quantities of $G_{k,d}$ in terms of the corresponding Riemannian quantities of $ST_{k,d}$ that are readily computable.

We avoid these details and directly use the friendly formulae provided in [2]. We will need to perform two major calculations to realize Algorithm 1.1. The first is to calculate the gradient and the second is to calculate the subspace obtained by walking along the geodesic defined by the gradient for a specific amount of time.

3.1.1 Computing the gradient

The Riemannian gradient of a function $F : G_{k,d} \to \mathbb{R}$ as such is a vector field which satisfies the identity,

$$\langle \nabla F, \xi \rangle_{\mathcal{W}} = \xi f, \quad \forall \xi \in T_{\mathcal{W}} G_{k,d}$$

To work with this quantity over the Grassmannian we will compute the horizontal lift of it over $ST_{k,d}$. Let $W \in ST_{k,d}$ and $\mathcal{W} \in G_{k,d}$ be s.t. $\pi(W) = \mathcal{W}$. Then for every $\xi \in T_{\mathcal{W}} G_{k,d}$ there exists a unique horizontal lift $\tilde{\xi} \in T_W ST_{k,d}$. Moreover, according to Proposition 2.25 of [73], any smooth vector field on $G_{k,d}$, like ∇F, has a unique horizontal lift $\widetilde{\nabla F}$ on $ST_{k,d}$ s.t. $d\pi_W \widetilde{\nabla F}_W = \nabla F_{\mathcal{W}}$. We will also need the function $\widetilde{F} : ST_{k,d} \to \mathbb{R}$ defined as $\widetilde{F}(W) = F(\pi(W))$.

Now, using formula (14) of [2], we can compute the horizontal lift of the gradient as

$$\widetilde{\nabla F}_W = (I - YY^T)\nabla \widetilde{F}(W) W^T W \tag{3.2}$$

where quantity $\nabla \widetilde{F}$ can be computed as is done in an Euclidean space. We are thus equipped to work with the gradients of our function.

3.1.2 Computing the geodesic

A geodesic on a manifold is equivalent to a straight line in Euclidean space. It is a curve $\gamma : [0, 1] \to G_{k,d}$ s.t. the "acceleration" of the curve, i.e., it's second derivative, at every point in its domain is zero. We use formula (19) from [2] which gives the formula of a geodesic over the Grassmannian. Let $W_0 \in ST_{k,d}$ represent the starting point of the geodesic $\mathcal{W}(t)$ and let the full singular value decomposition of $W_0(W_0^T W_0)^{-1/2}$ be $U\Sigma V^T$. Then we have,

$$\mathcal{W}(t) = \pi(W_0(W_0^T W_0)^{-1/2} V \cos \Sigma t + U \sin \Sigma t) \tag{3.3}$$

3.2 Formulae for Gradient descent

In this section, we compute the formulae for the gradient of F defined in (3.1) and the corresponding geodesic over which we walk to realize Algorithm 1.1. We will only work with matrices in $ST_{k,d}$ for which the columns are orthonormal, since this will greatly simplify all our formulae.

Lemma 3.1. *Let $Y \in ST_{k,d}$. Assume that f has unique minimum over the span of Y. Let $x_Y^* = \arg\min_x f(Yx)$. Then we have,*

$$\nabla \widetilde{F}_Y = \nabla f(Y x_Y^*) {x_Y^*}^T$$
$$\widetilde{\nabla F}_Y = \nabla f(Y x_Y^*) {x_Y^*}^T$$

Proof. Let $\delta E_{ij} \in \mathbb{R}^{d\times k}$ be s.t. for i, j, $\Delta Y_{ij} = \delta$ and it is equal to 0 in all other co-ordinates. Then we have

$$\begin{aligned}
\min_{x\in\mathbb{R}^k} f((Y+\Delta Y)x) &= f((Y+\Delta Y)x^*_{Y+\Delta Y}) \\
&= f((Y+\Delta Y)(x^*_Y + \Delta x^*_Y)) \\
&= f(Yx^*_Y + \Delta Y x^*_Y + Y\Delta x^*_Y + \Delta Y \Delta x^*_Y) \\
&= f(Yx^*_Y) + \nabla f(Yx^*_Y)^T(\Delta Y x^*_Y + Y\Delta x^*_Y + \Delta Y\Delta x^*_Y) + O(\delta^2) \\
&= f(Yx^*_Y) + \delta \nabla_i f(Yx^*_Y)(x^*_Y)_j + \nabla f(Yx^*_Y)^T Y \Delta x^*_Y + O(\delta^2) \\
&= f(Yx^*_Y) + \delta \nabla_i f(Yx^*_Y)(x^*_Y)_j + O(\delta^2)
\end{aligned}$$

where the last equality follows from the optimality of x^*_Y. Now

$$\begin{aligned}
\frac{\partial \widetilde{F}(Y)}{\partial Y_{ij}} &= \lim_{\delta\to 0} \frac{\min_{x\in\mathbb{R}^k} f((Y+\Delta Y)x) - \min_{x\in\mathbb{R}^k} f(Yx)}{\delta} \\
&= \lim_{\delta\to 0} \frac{f((Y+\Delta Y)x^*_{Y+\Delta Y}) - f(Yx^*_Y)}{\delta} \\
&= \lim_{\delta\to 0} \frac{f(Yx^*_Y) + \delta\nabla_i f(Yx^*_Y)(x^*_Y)_j + O(\delta^2) - f(Yx^*_Y)}{\delta} \\
&= \nabla_i f(Yx^*_Y)(x^*_Y)_j
\end{aligned}$$

which gives us the first desired result. Substituting this in (3.2) gives us the second desired result. □

Lemma 3.2. *The subspace $\mathcal{Y}'$ obtained after travelling for time t on the geodesic starting at the subspace $\mathcal{Y}$ with velocity $\nabla F_{\mathcal{Y}}$ is spanned by*

$$Y' = \left[Y\bar{x}^*_Y \cos(\Sigma_{11}t) + \frac{\nabla f(Yx^*_Y)}{\|\nabla f(Yx^*_Y)\|}\sin(\Sigma_{11}t) \quad YV' \right]$$

where $\bar{x}^*_Y$ is a unit vector in the direction of x^*_Y, $V' \in \mathbb{R}^{k\times k-1}$ is a basis for the space orthogonal to $\bar{x}^*_Y$ in $\mathbb{R}^k$ and $\Sigma_{11} = \|x^*_Y\|\|\nabla f(Yx^*_Y)\|$.

Proof. Let $Y \in ST_{k,d}$ have orthonormal columns that span $\mathcal{Y}$. From Lemma 3.1 we have,

$$\widetilde{\nabla F_Y} = \nabla f(Yx^*_Y){x^*_Y}^T$$

Now, to compute the point at distance t on the geodesic we use (3.3). Let $U\Sigma V^T$ be the compact SVD of $\widetilde{\nabla F_Y}$, then:

$$V = [\bar{x}^*_Y \quad V'], \quad \Sigma_{11} = \|x^*_Y\|\|\nabla f(Yx^*_Y)\|, \quad \Sigma_{ii} = 0 \text{ for } i \neq 1$$

$$\text{and} \quad U_1 = \frac{-\nabla f(Yx^*_Y)}{\|\nabla f(Yx^*_Y)\|}$$

these can be plugged in to directly obtain the given formula. □

3.3 Convergence Result

In this section, we present our convergence result in Theorem 3.1. Note that we get the same rate of convergence as does the ordinary gradient descent on Euclidean space.

Theorem 3.1. *For a convex function f with L-Lipschitz gradients, gradient descent over $G_{k,d}$ with a suitable step size gives a subspace $\mathcal{Y}_i$ s.t. starting from $\mathcal{Y}_0$ we have*

$$f(Y_i x^*_{Y_i}) - f(y^*) \leq \frac{\|Y_0 x^*_{Y_0} - y^*\|^2}{2iL}$$

where y^* is the global solution. Hence, in $O\left(\frac{1}{\epsilon}\right)$ steps we can find a $\mathcal{Y}$ s.t. $f(Yx^*_Y) - f(y^*) \leq \epsilon$.

Proof. For a suitably chosen $t > 0$, from Lemma 3.2 the subspace $\mathcal{Y}'$ at iteration $i+1$ is spanned by

$$Y' = \left[Y\bar{x}^*_Y \cos(\Sigma_{11} t) + \frac{\nabla f(Yx^*_Y)}{\|\nabla f(Yx^*_Y)\|} \sin(\Sigma_{11} t) \quad YV' \right]$$

We will now bound $\min_{y \in \mathcal{Y}'} f(y)$ as follows: for any $y_1, y_2 \in \mathbb{R}^d$, by Taylor's series expansion we know that there exists a $z \in \mathbb{R}^d$ s.t.

$$f(y_1) = f(y_2) + \nabla f(y_2)^T (y_1 - y_2) + \frac{1}{2}(y_1 - y_2)^T \nabla^2 f(y_2)(y_1 - y_2)$$

Since f has L-Lipschitz gradients $\nabla^2 f(z) \preceq LI$ and we get

$$f(y_1) \leq f(y_2) + \nabla f(y_2)^T (y_1 - y_2) + \frac{L}{2}\|y_1 - y_2\|^2 \tag{3.4}$$

On choosing the following:

$$y_1 = Yx^*_Y + \frac{\nabla f(Yx^*_Y)}{\|\nabla f(Yx^*_Y)\|} \|x^*_Y\| \tan(\Sigma_{11} t) \quad \text{and} \quad y_2 = Yx^*_Y$$

and noting that $y_1 \in \mathcal{Y}'$ we get

$$\min_{y \in \mathcal{Y}'} f(y) \leq f(y_1) \leq f(y_2) + \|\nabla f(y_2)\| \|x^*_Y\| \tan(\Sigma_{11} t) + \frac{L}{2}\|x^*_Y\|^2 \tan^2(\Sigma_{11} t)$$

On setting $\tan(\Sigma_{11} t) = -\dfrac{\|\nabla f(y_2)\|}{L\|x^*_Y\|}$ we get

$$f(Y' x^*_{Y'}) \leq f(Yx^*_Y) - \frac{\|\nabla f(Yx^*_Y)\|^2}{2L} \tag{3.5}$$

From here the proof follows exactly as in the case of gradient descent (see (6.3) in [1] and the proof thereafter) to give after i iterations a subspace $\mathcal{Y}_i$ s.t. starting from $\mathcal{Y}_0$ we have

$$f(Y_i x^*_{Y_i}) - f(y^*) \leq \frac{\|Y_0 x^*_{Y_0} - y^*\|^2}{2iL}$$

where y^* is the global solution. □

4 | Euclidean optimization on the Multinomial manifold

In the previous chapter, we formulated Euclidean optimization over the Grassmannian. In situations where the function f is defined using a data matrix $A \in \mathbb{R}^{n\times d}$ the technique effectively reduces the column dimension of the data matrix from d to k for the smaller sub-problems that it uses the black box for. For example, in the case of linear regression, the parameters $x \in \mathbb{R}^d$ are multiplied with A to get Ax. When we restrict this problem to a subspace defined by the columns of a matrix $W \in \mathbb{R}^{d\times k}$, we essentially only work with x of the form Wy for $y \in \mathbb{R}^k$. This means that the smaller-dimensional problem is defined by the matrix AW whose column dimension now is k.

In this chapter, we develop a different technique which can be used to reduce the row dimension of the data matrix for a certain class of optimization problems. We formulate this by taking convex combinations of the rows of matrix A. This requires using the Multinomial manifold $P_{m,n}$ which is defined as follows:

$$P_{m,n} = \{Y \in \mathbb{R}^{n\times m} : Y > 0 \text{ and } \forall j \in [m], \sum_i Y_{ij} = 1\}$$

Setup. Let $A \in \mathbb{R}^{n\times d}$ be the data matrix, $x \in \mathbb{R}^d$, $f : \mathbb{R}^d \to \mathbb{R}, g : \mathbb{R}^n \to \mathbb{R}$ and $\phi : \mathbb{R} \to \mathbb{R}$ be functions s.t.

$$f(x) = g(Ax) = \sum_{i=1}^{n} \phi(a_i^T x)$$

where a_i is the i-th row of A. Many popular machine learning models like logistic regression and more broadly generalized linear models can be written in this form.

For $Y \in P_{m,n}$ consider the matrix $Y^T A$. The rows of this matrix are convex combinations of the rows of A. To define a problem using a smaller row dimension data matrix, we define a function $\widetilde{f} : \mathbb{R}^d \to \mathbb{R}$ and $\widetilde{g} : \mathbb{R}^m \to \mathbb{R}$ as

$$\widetilde{f}(x) = \widetilde{g}(Y^T Ax) = \sum_{i=1}^{m} \phi((Y^T Ax)_i)$$

Let x_Y^* be the minimizer of $\widetilde{f}$. Then $f(x_Y^*)$ quantifies how well the solution obtained for $\widetilde{f}$ works for f. This gives us a way of defining an optimization problem over the multinomial manifold. Define $F : P_{m,n} \to \mathbb{R}$ as follows:

$$F(Y) = g(Ax_Y^*), \quad \text{where } x_Y^* = \arg\min_{x\in\mathbb{R}^d} \widetilde{g}(Y^T Ax) \tag{4.1}$$

Since $P_{m,n}$ can be endowed with the structured of a Riemannian manifold, we can use Algorithm 1.1 to optimize F. To flesh out the details of this technique we first describe the differential geoemtry of $P_{m,n}$ in Section 4.1, we then provide the formulae for the various Riemannian quantities in Section 4.2 followed by a convergence result in Section 4.3.

4.1 Differential geometry of the Multinomial manifold

We borrow the differential geometric treatment of $P_{m,n}$, including the gradient and retraction formulae as presented later, from [97]. In this work, $P_{m,n}$ is treated as an embedded Riemannian manifold of $\mathbb{R}^{n\times m}$, equipped with the Fisher information metric defined as:

$$g_U(\xi_U, \eta_U) = \sum_{i=1}^{n}\sum_{j=1}^{m} \frac{(\xi_U)_{ij}(\eta_U)_{ij}}{U_{ij}}, \quad \forall U \in P_{m,n} \text{ and } \forall \xi_U, \eta_U \in T_U P_{m,n}$$

4.1.1 Computing the gradient

Since we realize $P_{m,n}$ as an embedded submanifold of $\mathbb{R}^{n\times m}$, we can use the formulae for the Euclidean gradient of a function and "project" it down to the manifold using the following orthogonal projector w.r.t. the Fisher metric defined above. For $Y \in P_{m,n}$, define the linear function $\Pi_Y : \mathbb{R}^{n\times m} \to T_Y P_{m,n}$ as

$$\Pi_Y(Z) = Z - (1_n 1_n^T Z) \odot Y$$

where $1_n \in \mathbb{R}^n$ is a vector with all entries as ones and $\odot$ is the Hadamard product (entry-wise multiplication of matrices).

Now, to compute the gradient, define $\bar{F} : \mathbb{R}^{n\times m} \to \mathbb{R}$, the extension of F from $P_{m,n}$ to its ambient space $\mathbb{R}^{n\times m}$ as follows

$$\bar{F}(Z) = g(Ax_Z^*), \quad \text{where } x_Z^* = \arg\min_{x\in\mathbb{R}^d} \widetilde{g}(Z^T Ax)$$

We then have the following formula for the gradient of F in terms of the gradient of $\bar{F}$:

$$\nabla F_Y = \Pi_Y \left(\left.\frac{\partial \bar{F}}{\partial Z}\right|_Y \odot Y \right) \tag{4.2}$$

4.1.2 Computing the retraction

In the previous chapter, to move on the Grassmannian we used a geodesic. But to move on $P_{m,n}$ we instead use retractions. Retractions are mappings from the tangent space of a manifold to the manifold itself. In the case of $P_{m,n}$ they are easier to compute than the geodesics, so we prefer them.

At each point $Y \in P_{m,n}$ we define the retraction $R_Y : T_Y P_{m,n} \to P_{m,n}$, for $\xi_Y \in T_Y P_{m,n}$, as follows:

$$R_Y(\xi_Y) = (Y \odot \exp(\xi_Y \oslash Y)) \oslash (1_n 1_n^T (Y \odot \exp(\xi_Y \oslash Y))) \tag{4.3}$$

where $\oslash$ is the Hadamard division (entry-wise division of matrices) and $\exp : \mathbb{R}^{n\times m} \to \mathbb{R}^{n\times m}$ is the element-wise exponentiaion of matrices.

In our setting, if we move for time t in the direction ∇F_Y, we will use the retraction of $t \cdot \nabla F_Y$ on the manifold. Note that in the case where ξ_Y is of the type of ∇F_Y we can simplify the retraction formula to the following:

$$R_Y(\nabla F_Y) = \left(Y \odot \exp\left(\frac{\partial \bar{F}}{\partial Z}\bigg|_Y\right)\right) \oslash \left(1_n 1_n^T \left(Y \odot \exp\left(\frac{\partial \bar{F}}{\partial Z}\bigg|_Y\right)\right)\right) \tag{4.4}$$

4.2 Formulae for gradient descent

In this section, we compute the formula for the gradient of function F defined in (4.1) in Lemma 4.1.

Lemma 4.1. *Let $B = Z^T A \in \mathbb{R}^{m\times d}$, then for the function $\bar{F}$ defined as above, we have*

$$\begin{aligned}\frac{\partial \bar{F}(Z)}{\partial Z} = Ax_Z^* \nabla g(Ax_Z^*)^T A \left[B^T \nabla^2 \widetilde{g}(Bx_Z^*) B\right]^{-1} B^T \nabla^2 \widetilde{g}(Bx_Z^*) + \\ A \left[B^T \nabla^2 \widetilde{g}(Bx_Z^*) B\right]^{-1} A^T \nabla g(Ax_Z^*) \nabla \widetilde{g}(Bx_Z^*)^T\end{aligned}$$

Remark 4.1. *Note that in the gradient expression of the above lemma, we have the gradient of the local function $\nabla \widetilde{g}(Bx_Z^*) \in \mathbb{R}^m$, the hessian of the local function $\nabla^2 \widetilde{g}(Bx_Z^*) \in \mathbb{R}^{m\times m}$ and the gradient of the global function $\nabla g(Ax_Z^*) \in \mathbb{R}^n$. Also, note that if B is a full rank matrix then the second term in the above expression should be zero because $\nabla \widetilde{g}(Bx_Z^*)^T$ will be zero.*

Proof. We have,

$$\frac{\partial \bar{F}(Z)}{\partial Z} = \frac{\partial g(Ax_Z^*)}{\partial Z} = \nabla g(Ax_Z^*)^T A \frac{\partial x_Z^*}{\partial Z} \tag{4.5}$$

To obtain the derivative of x_Z^* w.r.t. Z we use the inverse function theorem. Since x_Y^* is a minimizer of the function $\widetilde{f}$, we have $B^T \nabla \widetilde{g}(Bx_Y^*) = 0$. Define, $\Phi : \mathbb{R}^{n\times m} \times \mathbb{R}^d \to \mathbb{R}^d$ as

$$\Phi(Z, x) = B^T \nabla \widetilde{g}(Bx_Z^*)$$

and let $\Phi_l(Z, x) = \Phi(Z, x)_l$. Then,

$$\begin{aligned}\frac{\partial \Phi_l}{\partial x^i} &= \frac{\partial B_l^T \nabla \widetilde{g}(Bx)}{\partial x^i} \\ &= B_{pl} \frac{\partial (\nabla \widetilde{g}(Bx))_p}{\partial x^i} \\ &= B_{pl} (\nabla^2 \widetilde{g}(Bx))_{pq} \frac{\partial (Bx)_q}{\partial x^i} \\ &= B_{pl} (\nabla^2 \widetilde{g}(Bx))_{pq} B_{qi} \\ &= B_l^T \nabla^2 \widetilde{g}(Bx) B_i\end{aligned}$$

and,

$$\begin{aligned}
\frac{\partial \Phi_l}{\partial Z^{ij}} &= \frac{\partial B_l^T \nabla \widetilde{g}(Bx)}{\partial Z^{ij}} \\
&= B_{pl} \frac{\partial (\nabla \widetilde{g}(Bx))_p}{\partial Z^{ij}} + \left(\frac{\partial B_{pl}}{\partial Z^{ij}} \right) (\nabla \widetilde{g}(Bx))_p \\
&= B_{pl} (\nabla^2 \widetilde{g}(Bx))_{pq} \frac{\partial (Bx)_q}{\partial Z^{ij}} + \left(\frac{\partial (Z_p^T A_l)}{\partial Z^{ij}} \right) (\nabla \widetilde{g}(Bx))_p \\
&= B_{pl} (\nabla^2 \widetilde{g}(Bx))_{pq} \frac{\partial (Z_q^T Ax)}{\partial Z^{ij}} + A_{il} (\nabla \widetilde{g}(Bx))_j \\
&= B_{pl} (\nabla^2 \widetilde{g}(Bx))_{pj} (Ax)_i + A_{il} (\nabla \widetilde{g}(Bx))_j \\
&= B_l^T (\nabla^2 \widetilde{g}(Bx))_j (Ax)_i + A_{il} (\nabla \widetilde{g}(Bx))_j
\end{aligned}$$

Hence,

$$\begin{aligned}
\frac{\partial x_Z^*}{\partial Y^{ij}} &= \left[\frac{\partial \Phi_l}{\partial x^p} \right]^{-1} \left[\frac{\partial \Phi_l}{\partial Z^{ij}} \right] \\
&= \left[B^T \nabla^2 \widetilde{g}(Bx) B \right]^{-1} \left[B^T (\nabla^2 \widetilde{g}(Bx))_j (Ax)_i + a_i (\nabla \widetilde{g}(Bx))_j \right]
\end{aligned}$$

where a_i is the i-th row of A. Substituting this in (4.5), we get the desired result. □

4.3 Convergence result

Now we will show that a line search-based gradient descent method on $P_{m,n}$ will converge to the solution of a convex function f. In Theorem 4.2 below, we show that if Y is a critical point of F then x_Y^* is a critical point of f. Using Theorem 4.3.1 from [3] (which we state below as Theorem 4.1 without proof), we know that the line search algorithm will converge to a critical point of F. Hence, if f has only one critical point which is its minima, then Algorithm 1.1 converges to this minima.

Theorem 4.1 ([3])**.** *Let $\{Y_k\}$ be an infinite sequence of iterates generated by Accelerated Line Search. Then every accumulation point of $\{Y_k\}$ is a critical point of the function F.*

Theorem 4.2. *Assume that for any $y \in \mathbb{R}^m$ s.t. all coordinates of y are equal, y is not a critical point of the local problem $\widetilde{g}(y)$. Also, assume that all critical points of $\widetilde{f}$ are either strict maximas or minimas, i.e., the hessians at these points are invertible. Then for $Y \in P_{m,n}$, a critical point of F, x_Y^* is a critical point of f.*

Proof. Since Y is a critical point of F, we have $\nabla F_Y = 0$. Now,

$$\begin{aligned}
\nabla F_Y &= \Pi_Y \left(\left. \frac{\partial \bar{F}}{\partial Z} \right|_Y \odot Y \right) \\
&= \left. \frac{\partial \bar{F}}{\partial Z} \right|_Y \odot Y - \left(1_n 1_n^T \left(\left. \frac{\partial \bar{F}}{\partial Z} \right|_Y \odot Y \right) \right) \odot Y \\
&= \left(\left. \frac{\partial \bar{F}}{\partial Z} \right|_Y - \left(1_n 1_n^T \left(\left. \frac{\partial \bar{F}}{\partial Z} \right|_Y \odot Y \right) \right) \right) \odot Y
\end{aligned}$$

Since $\forall\ i \in [n], j \in [m],\ Y_{ij} > 0$, to satisfy the critical point condition we have,

$$\left(\frac{\partial \bar{F}}{\partial Z}\Big|_Y - \left(1_n 1_n^T \left(\frac{\partial \bar{F}}{\partial Z}\Big|_Y \odot Y\right)\right)\right)_{ij} = 0 \implies \left(\frac{\partial \bar{F}}{\partial Z}\Big|_Y\right)_{ij} = \left(\frac{\partial \bar{F}}{\partial Z}\Big|_Y\right)_j^T Y_j$$

Since this is true for each i, j, we conclude that for each j, all the co-ordinates of $\left(\frac{\partial \bar{F}}{\partial Z}\big|_Y\right)_j$ are equal. Which in turn implies that $\frac{\partial \bar{F}}{\partial Z}\big|_Y$ is a rank 1 or a rank 0 matrix.

Now, we will show that it cannot be a rank 1 matrix. We have,

$$\frac{\partial \bar{F}}{\partial Z}\Big|_Y = Ax_Y^* \nabla g(Ax_Y^*)^T A \left[B^T \nabla^2 \widetilde{g}(Bx_Y^*)B\right]^{-1} B^T \nabla^2 \widetilde{g}(Bx_Y^*) + A \left[B^T \nabla^2 \widetilde{g}(Bx_Y^*)B\right]^{-1} A^T \nabla g(Ax_Y^*) \nabla \widetilde{g}(Bx_Y^*)^T$$

This is a matrix of the form $u_1 v_1^T + u_2 v_2^T$ where $u_1, u_2 \in \mathbb{R}^n$ and $v_1, v_2 \in \mathbb{R}^m$. We consider two cases:

1. $A^T \nabla g(Ax_Y^*) = 0$: This implies that it's a rank 0 matrix.
2. $A^T \nabla g(Ax_Y^*) \neq 0$: Since the hessian of the local problem $\widetilde{f}$ is invertible at x_Y^* we have that $\nabla^2 \widetilde{f}(x_Y^*) = B^T \nabla^2 \widetilde{g}(Bx_Y^*)B$ is an invertible matrix which in turn implies that $B^T \nabla^2 \widetilde{g}(Bx_Y^*)$ is a full rank matrix. Hence, $\left[B^T \nabla^2 \widetilde{g}(Bx_Y^*)B\right]^{-1} A^T \nabla g(Ax_Y^*) \neq 0$ and $v_1 \neq 0$. Now we consider the following sub-cases:
   (a) $u_2 = 0$: This implies that Ax_Y^* has all the coordinates to be equal, which in turn implies that Bx_Y^* has all co-ordinates to be equal. But such a point cannot be a critical point from the assumption.
   (b) $u_1 = 0$: This implies that $Bx_Y^* = Y^T Ax_Y^* = 0$, but this cannot be a critical point of $\widetilde{g}$ from the assumption.
   (c) $v_2 = 0$: This too implies that Ax_Y^* has all the coordinates to be equal.
   (d) $u_1 \parallel u_2$: This also implies that Ax_Y^* has all coordinates equal.
   (e) $v_1 \parallel v_2$: This implies that $v_1^T B \parallel v_2^T B$. But $v_1^T B = \nabla g(Ax_Y^*)^T A$ and $v_2^T B = 0$, the latter following from the fact that x_Y^* is a critical point of $\widetilde{f}$. This is a direct contradiction to the assumption.

Hence, $\frac{\partial \bar{F}}{\partial Z}\big|_Y$ is a rank 0 matrix.

Now, we show that if the matrix is rank 0 then $A^T \nabla g(Ax_Y^*) = 0$, establishing that x_Y^* is a critical point of f. From the arguments used above we know that $u_1 \neq 0$, hence $v_1 = 0$. But $\nabla^2 \widetilde{f}(x_Y^*) = B^T \nabla^2 \widetilde{g}(Bx_Y^*)B$ is invertible and $B^T \nabla^2 \widetilde{g}(Bx_Y^*)$ is a full rank matrix. Hence, for $v_2 = 0$, we need $A^T \nabla g(Ax_Y^*) = 0$. □

Part III

Minimax estimators using online learning

5 | Learning minimax estimators

Estimating the properties of a probability distribution is a fundamental problem in machine learning and statistics. In this problem, we are given observations generated from an unknown probability distribution P belonging to a class of distributions $\mathcal{P}$. Knowing $\mathcal{P}$, we are required to estimate certain properties of the unknown distribution P, based on the observations. Designing good and "optimal" estimators for such problems has been a fundamental subject of research in statistics. Over the years, statisticians have considered various notions of optimality to compare the performance of estimators and to aid their search of good estimators. Some popular notions of optimality include admissibility, minimax optimality, Bayesian optimality, asymptotic efficiency [34, 74]. Of these, minimax optimality is the most popular notion and has received wide attention in frequentist statistics. This notion of optimality has led to the minimax estimation principle, where the goal is to design estimators with the minimum worst-case risk. Let $R(\hat{\theta}, \theta(P))$ be the risk of an estimator $\hat{\theta}$ for estimating the property $\theta(P)$ of a distribution P, where an estimator is a function which maps observations to the set of possible values of the property. Then the worst-case risk of $\hat{\theta}$ is defined as $\sup_{P \in \mathcal{P}} R(\hat{\theta}, \theta(P))$. The goal in minimax estimation principle is to design estimators with worst-case risk close to the best worst-case risk, which is defined as $R^* = \inf_{\hat{\theta}} \sup_{P \in \mathcal{P}} R(\hat{\theta}, \theta(P))$, where the infimum is computed over the set of all estimators. Such estimators are often referred to as minimax estimators [99].

Classical Estimators. A rich body of work in statistics has focused on studying the minimax optimality properties of classical estimators such as the maximum likelihood estimator (MLE), Bayes estimators, and minimum contrast estimators (MCEs) [14, 15, 50, 70, 101, 109]. Early works in this line have considered parametric estimation problems and focused on the asymptotic setting, where the number of observations approaches infinity, for a fixed problem dimension. In a series of influential works, Hájek and Le Cam showed that under certain regularity conditions on the parametric estimation problem, MLE is asymptotically minimax whenever the risk is measured with respect to a convex loss function [50, 70]. Later works in this line have considered both parametric and non-parametric estimation problems in the non-asymptotic setting and studied the minimax rates of estimation. In a series of works, Birgé [14, 15] showed that under certain regularity conditions on the model class $\mathcal{P}$ and the estimation problem, MLE and MCEs are approximately minimax w.r.t Hellinger distance.

While these results paint a compelling picture of classical estimators, we highlight two key problem settings where they tend to be rate inefficient (that is, achieve sub-optimal worst-case risk) [15, 105]. The first is the so-called high dimensional sampling setting,

where the number of observations is comparable to the problem dimension, and under which, classical estimators can be highly sub-optimal. In some recent work, Jiao et al. [53] considered the problem of entropy estimation in discrete distributions and showed that the MLE (plug-in rule) is sub-optimal in the high dimensional regime. Similarly, Cai and Low [19] considered the problem of estimation of non-smooth functional $\frac{1}{d}\sum_{i=1}^{d}|\theta_i|$ from an observation $Y \sim \mathcal{N}(\theta, I_d)$ and showed that the MLE is sub-optimal. The second key setting where classical estimators tend to be sub-optimal is when the risk $R(\hat{\theta}, \theta(P))$ is measured w.r.t "non-standard" losses that have a very different behavior compared to standard losses such as Kullback-Leibler (KL) divergence. For example, consider the MLE, which can be viewed as a KL projection of the empirical distribution of observations onto the class of distributions $\mathcal{P}$. By its design, we expect it to be minimax when the risk is measured w.r.t KL divergence and other related metrics such as Hellinger distance [15]. However, for loss metrics which are not aligned with KL, one can design estimators with better performance than MLE, by taking the loss into consideration. This phenomenon is better illustrated with the following toy example. Suppose $\mathcal{P}$ is the set of multivariate normal distributions in $\mathbb{R}^d$ with identity covariance, and suppose our goal is to estimate the mean of a distribution $P \in \mathcal{P}$, given n observations drawn from it. If the risk of estimating θ as $\tilde{\theta}$ is measured w.r.t the following loss $\|\tilde{\theta} - \theta - c\|_2^2$, for some constant c, then it is easy to see that MLE has a worst-case risk greater than $\|c\|_2^2$. Whereas, the minimax risk R^* is equal to d/n, which is achieved by an estimator obtained by shifting the MLE by c. While the above loss is unnatural, such a phenomenon can be observed with natural losses such as ℓ_q norms for $q \in (0, 1)$ and asymmetric losses.

Bespoke Minimax Estimators. For problems where classical estimators are not optimal, designing a minimax estimator can be challenging. Numerous works in the literature have attempted to design minimax estimators in such cases. However the focus of these works is on specific problems [18, 19, 53, 102], and there is no single estimator which is known to be optimal for a wide range of estimation problems. For example, Jiao et al. [53], Wu and Yang [108] considered the problem of entropy estimation for discrete distributions and provided a minimax estimator in the high-dimensional setting. Cai and Low [19] considered the problem of estimating a non-smooth functional in high dimensions and provided a minimax estimator. While these results are impressive, the techniques used in these works are tailored towards specific problems and do not extend to other problems. So, a natural question that arises in this context is, how should one go about constructing minimax estimators for problems where none of the classical estimators are optimal? Unfortunately, our current understanding of minimax estimators does not provide any concrete guidelines on designing such estimators.

Minimax Estimation via Solving Statistical Games. In this work, we attempt to tackle the problem of designing minimax estimators from a game-theoretic perspective. Instead of the usual two-step approach of first designing an estimator and then certifying its minimax optimality, we take a more direct approach and attempt to directly solve the following min-max statistical game: $\inf_{\hat{\theta}} \sup_{P \in \mathcal{P}} R(\hat{\theta}, \theta(P))$. Since the resulting estimators are solutions to the min-max game, they are optimal by construction. Such a direct approach for construction of minimax estimators has certain advantages over the classical estimators. First, the technique itself is very general and can *theoretically* be used to construct minimax estimators for any estimation problem. Second, a direct approach often results in *exact* minimax estimators with $R^* + o(1)$ worst-case risk. In contrast, classical

estimators typically achieve $O(1)R^*$ worst-case risk, which is constant factors worse than the direct approach. Finally, a direct approach can make effective use of any available side information about the problem, to construct estimators with better worst-case risk than classical estimators. For example, consider the problem of mean estimation given samples drawn from an unknown Gaussian distribution. If it is known a priori that the true mean lies in a bounded set, then a direct approach for solving the min-max statistical game results in estimators with better performance than classical estimators. Several past works have attempted to directly solve the min-max game associated with the estimation problem [see 9, and references therein]. We discuss these further in Section 5.1 after providing some background, but in gist, existing approaches either focus on specific problems or are applicable only to simple estimation problems.

This Work. In this work, we rely on recent advances in online learning and game theory to directly solve the min-max statistical game. Recently, online learning techniques have been widely used for solving min-max games. For example, Freund and Schapire [37] relied on these techniques to find equilibria in min-max games that arise in the context of boosting. Similar techniques have been explored for robust optimization by Chen et al. [23], Feige et al. [32]. In this work, we take a similar approach and provide an algorithm for solving statistical games. A critical distinction of statistical games, in contrast to the typical min-max games studied in the learning and games literature, is that the domain of all possible measurable estimators is extremely large, the set of possible parameters need not be convex, and the loss function need not be convex-concave. We show that it is nonetheless possible to finesse these technical caveats and solve the statistical game, provided we are given access to two subroutines: a Bayes estimator subroutine which outputs a Bayes estimator corresponding to any given prior, and a subroutine which computes the worst-case risk of any given estimator. Given access to these two subroutines, we show that our algorithm outputs both a minimax estimator and a least favorable prior. The minimax estimator output by our algorithm is a randomized estimator which is an ensemble of multiple Bayes estimators. When the loss function is convex - which is the case for a number of commonly used loss functions - the randomized estimator can be transformed into a deterministic minimax estimator. For problems where the two subroutines are efficiently implementable, our algorithm provides an efficient technique to construct minimax estimators. While implementing the subroutines can be computationally hard in general, we show that the computational complexity can be significantly reduced for a wide range of problems satisfying certain invariance properties.

To demonstrate the power of this technique, we use it to construct provably minimax estimators for the classical problems of finite dimensional Gaussian sequence model and linear regression. In the problem of Gaussian sequence model, we are given a single sample drawn from a normal distribution with mean θ and identity covariance, where $\theta \in \mathbb{R}^d, \|\theta\|_2 \leq B$. Our goal is to estimate θ well under squared-error loss. This problem has received much attention in statistics because of its simplicity and connections to non-parametric regression [55]. Surprisingly, however, the exact minimax estimator is unknown for the case when $B \geq 1.16\sqrt{d}$ [10, 12, 78]. In this work, we show that our technique can be used to construct provably minimax estimators for this problem, for general B. To further demonstrate that our technique is widely applicable, we present empirical evidence showing that our algorithm can be used to construct estimators for covariance and entropy estimation which match the performance of existing minimax estimators.

Outline. We conclude this section with a brief outline of the rest of the paper. We already provided the necessary background on minimax estimation and online learning and in Section 1.1.3 and 1.1.4 respectively. In Section 5.1, we introduce our algorithm for solving statistical games. In Sections 5.2, 5.3, 5.4 we utilize our algorithm to construct provably minimax estimators for finite dimensional Gaussian sequence model and linear regression. In Section 5.7 we study the empirical performance of our algorithm on a variety of statistical estimation problems. We defer technical details to Section 5.8.12.2. Finally, we refer the reader to Section 6.3 in Chapter 6 for a discussion of future directions and some open problems.

5.1 Minimax Estimation via Online Learning

In this section, we present our algorithm for computing a mixed strategy NE of the statistical game in Equation (1.1) (equivalently a pure strategy NE of the linearized game in Equation (1.4)). A popular and widely used approach for solving min-max games is to rely on online learning algorithms [21, 48]. In this approach, the minimization player and the maximization player play a repeated game against each other. Both the players rely on online learning algorithms to choose their actions in each round of the game, with the objective of minimizing their respective regret. The following proposition shows that this repeated game play converges to a NE.

Proposition 5.1. *Consider a repeated game between the minimization and maximization players in Equation (1.4). Let $(\hat{\theta}_t, P_t)$ be the actions chosen by the players in iteration t. Suppose the actions are such that the regret of each player satisfies*

$$\sum_{t=1}^{T} R(\hat{\theta}_t, P_t) - \inf_{\hat{\theta} \in \mathcal{D}} \sum_{t=1}^{T} R(\hat{\theta}, P_t) \leq \epsilon_1(T),$$
$$\sup_{\theta \in \Theta} \sum_{t=1}^{T} R(\hat{\theta}_t, \theta) - \sum_{t=1}^{T} R(\hat{\theta}_t, P_t) \leq \epsilon_2(T).$$

Let $\hat{\theta}_{RND}$ denote the randomized estimator obtained by uniformly sampling an estimator from the iterates $\{\hat{\theta}_t\}_{t=1}^T$. Define the mixture distribution P_{AVG} as $\frac{1}{T}\sum_{i=1}^{T} P_i$. Then $(\hat{\theta}_{RND}, P_{AVG})$ is an approximate mixed strategy NE of Equation (1.1)

$$R(\hat{\theta}_{RND}, P_{AVG}) \leq \inf_{\hat{\theta} \in \mathcal{D}} R(\hat{\theta}, P_{AVG}) + \frac{\epsilon_1(T) + \epsilon_2(T)}{T},$$
$$R(\hat{\theta}_{RND}, P_{AVG}) \geq \sup_{\theta \in \Theta} R(\hat{\theta}_{RND}, \theta) - \frac{\epsilon_1(T) + \epsilon_2(T)}{T}.$$

Note that the above proposition doesn't specify an algorithm to generate the iterates $(\hat{\theta}_t, P_t)$. All it shows is that as long as both the players rely on algorithms which guarantee sub-linear regret, the iterates converge to a NE. As discussed in Section 1.1.4, there exist several algorithms such as FTRL, FTPL, Best Response (BR), which guarantee sub-linear regret. It is important to choose these algorithms appropriately as our choices impact the rate of convergence to a NE and also the computational complexity of the resulting algorithm. First, consider the minimization player, whose domain $\mathcal{M}_{\mathcal{D}}$ is the set of all probability measures over $\mathcal{D}$. Note that $\mathcal{D}$, the set of all deterministic estimators, is an infinite dimensional

space. So, algorithms such as FTRL, FTPL, whose regret bounds depend on the dimension of the domain, can not guarantee sub-linear regret. So the minimization player is forced to rely on BR, which has 0 regret. Recall, in order to use BR, the minimization player requires the knowledge of the future action of the opponent. This can be made possible in the context of min-max games by letting the minimization player choose her action after the maximization player reveals her action. Next, consider the maximization player. Since the minimization player is relying on BR, the maximization player has to rely on either FTRL or FTPL to choose her action[1]. In this work we choose the FTPL algorithm studied by [96]. Our choice is mainly driven by the computational aspects of the algorithm. Each iteration of the FTRL algorithm of Krichene et al. [67] involves sampling from a general probability distribution. Whereas, each iteration of the FTPL algorithm requires minimization of a non-convex objective. While both sampling and optimization are computationally hard in general, the folklore is that optimization is relatively easier than sampling in many practical applications.

We now describe our algorithm for computing a pure strategy NE of Equation (1.4). In iteration t, the maximization player chooses distribution P_t using FTPL. P_t is given by the distribution of the random variable $\theta_t(\sigma)$, which is generated by first sampling a random vector $\sigma \in \mathbb{R}^d$ from exponential distribution and then computing an optimizer of

$$\sup_{\theta \in \Theta} \sum_{i=1}^{t-1} R(\hat{\theta}_i, \theta) + \langle \sigma, \theta \rangle \,. \tag{5.1}$$

The minimization player chooses $\hat{\theta}_t$ using BR, which involves computing a minimizer of the integrated risk under prior P_t

$$\inf_{\hat{\theta} \in \mathcal{D}} R(\hat{\theta}, P_t). \tag{5.2}$$

Very often, computing exact optimizers of the above problems is infeasible. Instead, one can only compute approximate optimizers. To capture the error from this approximation, we introduce the notion of approximate optimization oracles/subroutines.

Definition 5.1.1 (Maximization Oracle)**.** A function $\mathcal{O}^{\max}_{\alpha,\beta}(\cdot)$ is called (α, β)-approximate maximization oracle, if for any set of estimators $\{\hat{\theta}_i\}_{i=1}^T$ and perturbation σ, it returns $\theta' \in \Theta$ which satisfies the following inequality

$$\sum_{i=1}^{T} R\left(\theta', \theta\right) + \left\langle \sigma, \theta' \right\rangle \geq \sup_{\theta \in \Theta} \sum_{i=1}^{T} R(\hat{\theta}_i, \theta) + \langle \sigma, \theta \rangle - (\alpha + \beta \|\sigma\|_1) \,.$$

We denote the output θ' by $\mathcal{O}^{\max}_{\alpha,\beta}\left(\{\hat{\theta}_i\}_{i=1}^T, \sigma\right)$.

Definition 5.1.2 (Minimization Oracle)**.** A function $\mathcal{O}^{\min}_{\alpha}(\cdot)$ is called α-approximate minimization oracle, if for any probability measure P, it returns an approximate Bayes estimator $\hat{\theta}'$ which satisfies the following inequality

$$R(\hat{\theta}', P) \leq \inf_{\hat{\theta} \in \mathcal{D}} R(\hat{\theta}, P) + \alpha.$$

We denote the output $\hat{\theta}'$ by $\mathcal{O}^{\min}_{\alpha}(P)$.

[1]If both the players use BR, then both will wait for the other player to pick an action first. As a result, the algorithm will never proceed.

Algorithm 5.1 FTPL for statistical games

1: **Input:** Parameter of exponential distribution η, approximate optimization oracles $\mathcal{O}^{\max}_{\alpha,\beta}(\cdot), \mathcal{O}^{\min}_{\alpha'}(\cdot)$ for problems (5.1), (5.2) respectively
2: **for** $t = 1 \dots T$ **do**
3: Let P_t be the distribution of random variable $\theta_t(\sigma)$, which is generated as follows:
(i) Generate a random vector σ such that $\{\sigma_j\}_{j=1}^{d} \overset{i.i.d}{\sim} \text{Exp}(\eta)$
(ii) Compute $\theta_t(\sigma)$ as

$$\theta_t(\sigma) = \mathcal{O}^{\max}_{\alpha,\beta}\left(\{\hat{\theta}_i\}_{i=1}^{t-1}, \sigma\right).$$

4: Compute $\hat{\theta}_t$ as

$$\hat{\theta}_t = \mathcal{O}^{\min}_{\alpha'}(P_t).$$

5: **Output:** $\{\hat{\theta}_1, \dots \hat{\theta}_T\}, \{P_1, \dots P_T\}$.

Given access to subroutines $\mathcal{O}^{\max}_{\alpha,\beta}(\cdot), \mathcal{O}^{\min}_{\alpha'}(\cdot)$ for approximately solving the optimization problems in Equations (5.1), (5.2), the algorithm alternates between the maximization and minimization players who choose P_t and $\hat{\theta}_t$ in each iteration. We summarize the overall algorithm in Algorithm 5.1. The following theorem shows that Algorithm 5.1 converges to an approximate NE of the statistical game.

Theorem 5.1 (Approximate NE)**.** *Consider the statistical game in Equation (1.1). Suppose $\Theta \subseteq \mathbb{R}^d$ is compact with ℓ_∞ diameter D, i.e., $D = \sup_{\theta_1,\theta_2 \in \Theta} \|\theta_1 - \theta_2\|_\infty$. Suppose $R(\hat{\theta}, \theta)$ is L-Lipschitz in its second argument w.r.t ℓ_1 norm:*

$$\forall \hat{\theta}, \theta_1, \theta_2 \quad |R(\hat{\theta}, \theta_1) - R(\hat{\theta}, \theta_2)| \leq L\|\theta_1 - \theta_2\|_1.$$

Suppose Algorithm 5.1 is run for T iterations with approximate optimization subroutines $\mathcal{O}^{max}_{\alpha,\beta}(\cdot)$, $\mathcal{O}^{min}_{\alpha'}(\cdot)$ for solving the maximization and minimization problems. Let $\hat{\theta}_{RND}$ be the randomized estimator obtained by uniformly sampling an estimator from the iterates $\{\hat{\theta}_t\}_{t=1}^{T}$. Define the mixture distribution P_{AVG} as $\frac{1}{T}\sum_{i=1}^{T} P_i$. Then $(\hat{\theta}_{RND}, P_{AVG})$ is an approximate mixed strategy NE of the statistical game in Equation (1.1)

$$\sup_{\theta \in \Theta} R(\hat{\theta}_{RND}, \theta) - \epsilon \leq R(\hat{\theta}_{RND}, P_{AVG}) \leq \inf_{\hat{\theta} \in \mathcal{D}} R(\hat{\theta}, P_{AVG}) + \epsilon,$$

where $\epsilon = O\left(\eta d^2 + \frac{d(\beta T + D)}{\eta T} + \alpha + \alpha'\right)$.

As an immediate consequence of Theorem 5.1, we show that the minmax and maxmin values of the statistical game in Equation (1.4) are equal to each other. Moreover, when the risk is bounded, we show that the statistical game (1.1) has minimax estimators and LFPs.

Corollary 5.1 (Minimax Theorem)**.** *Consider the setting of Theorem 5.1. Then*

$$\inf_{\hat{\theta} \in \mathcal{M}_{\mathcal{D}}} \sup_{P \in \mathcal{M}_\Theta} R(\hat{\theta}, P) = \sup_{P \in \mathcal{M}_\Theta} \inf_{\hat{\theta} \in \mathcal{M}_{\mathcal{D}}} R(\hat{\theta}, P) =: R^*.$$

Furthermore, suppose the risk $R(\hat{\theta}, \theta)$ is a bounded function and Θ is compact w.r.t the following metric: $\Delta_M(\theta_1, \theta_2) = \sup_{\theta \in \Theta} |M(\theta_1, \theta) - M(\theta_2, \theta)|$. Then there exists a minimax estimator $\hat{\theta}^* \in \mathcal{M}_{\mathcal{D}}$ whose worst-case risk satisfies

$$\sup_{\theta \in \Theta} R(\hat{\theta}^*, \theta) = R^*,$$

and there exists a least favorable prior $P^* \in \mathcal{M}_\Theta$ *whose Bayes risk satisfies*

$$\inf_{\hat{\theta} \in \mathcal{D}} R(\hat{\theta}, P^*) = R^*.$$

We note that the assumption on compactness of Θ w.r.t Δ_M is mild and holds whenever Θ is compact w.r.t ℓ_2 norm and M is a continuous function. As another consequence of Theorem 5.1, we show that Algorithm 5.1 outputs approximate minimax estimators and LFPs.

Corollary 5.2. *Consider the setting of Theorem 5.1. Suppose Algorithm 5.1 is run with* $\eta = \sqrt{\frac{1}{dL^2T}}$. *Then the worst-case risk of* $\hat{\theta}_{RND}$ *satisfies*

$$\sup_{\theta \in \Theta} R(\hat{\theta}_{RND}, \theta) \leq R^* + O(d^{\frac{3}{2}} L T^{-\frac{1}{2}} + \alpha + \alpha' + \beta d^{\frac{3}{2}} L T^{\frac{1}{2}}).$$

Moreover, P_{AVG} *is approximately least favorable with the associated Bayes risk satisfying*

$$\inf_{\hat{\theta} \in \mathcal{D}} R(\hat{\theta}, P_{AVG}) \geq R^* - O(d^{\frac{3}{2}} L T^{-\frac{1}{2}} + \alpha + \alpha' + \beta d^{\frac{3}{2}} L T^{\frac{1}{2}}).$$

In addition, suppose the loss M *used in the computation of risk is convex in its first argument. Let* $\hat{\theta}_{AVG}$ *be the deterministic estimator which is equal to the mean of the probability distribution associated with* $\hat{\theta}_{RND}$. *Then the worst-case risk of* $\hat{\theta}_{AVG}$ *satisfies*

$$\sup_{\theta \in \Theta} R(\hat{\theta}_{AVG}, \theta) \leq R^* + O(d^{\frac{3}{2}} L T^{-\frac{1}{2}} + \alpha + \alpha' + \beta d^{\frac{3}{2}} L T^{\frac{1}{2}}),$$

and $\hat{\theta}_{AVG}$ *is an approximate Bayes estimator for prior* P_{AVG}.

Remark 5.1 (Near Optimal Estimator)**.** *Corollary 5.2 shows that when the approximation error of the optimization oracles is sufficiently small and when* T *is large enough, Algorithm 5.1 outputs a minimax estimator with worst-case risk* $(1 + o(1))R^*$. *This improves upon the approximate minimax estimators that are usually designed in statistics, which have a worst-case risk of* $O(1)R^*$.

Remark 5.2 (Deterministic Minimax Estimators)**.** *For general non-convex loss functions, Algorithm 5.1 only provides a randomized minimax estimator. Given this, a natural question that arises is whether there exist efficient algorithms for finding a deterministic minimax estimator. Unfortunately, even with access to the optimization subroutines used by Algorithm 5.1, finding a deterministic minimax estimator can be NP-hard [see Theorem 9 of 23]*

Remark 5.3 (Implementation Details)**.** *Note that the estimators* $\{\hat{\theta}_i\}_{i=1}^T$ *and distributions* $\{P_i\}_{i=1}^T$ *output by Algorithm 5.1 are infinite dimensional objects and can not in general be stored using finitely many bits. However, in practice, we use independent samples generated from* P_i *as its proxy and only work with these samples. Since* $\hat{\theta}_i$ *is a Bayes estimator for prior* P_i, *it can be approximately computed using samples from* P_i. *This process of approximating* P_i *with its samples introduces some approximation error and the number of samples used in this approximation need to be large enough to ensure Algorithm 5.1 returns a minimax estimator. For the problems of finite Gaussian sequence model and linear regression studied in Sections 5.3, 5.4, we show that poly*(d) *samples suffice to ensure a minimax estimator.*

Remark 5.4 (Computation of the Oracles)**.** *We now consider the computational aspects involved in the implementation of optimization oracles used by Algorithm 5.1. Recall that the maximization oracle, given any estimator, computes its worst-case risk with some linear perturbation. Since this objective could potentially be non-concave, maximizing it can take exponential time in the worst-case. And recall that the minimization oracle computes the Bayes estimator given some prior distribution. Implementation of this minimization oracle can also be computationally expensive in the worst case. While the worst case complexities are prohibitive, for a number of problems, one can make use of* the problem structure *to efficiently implement these oracles in polynomial time.*

In particular, we leverage symmetry and invariance properties of the statistical games to reduce the complexity of optimization oracles, while controlling their approximation errors; see Section 5.2. We further consider the case where there is no structure in the problem, other than the existence of finite-dimensional sufficient statistics for the statistical model. This allows one to reduce the computational complexity of the minimization oracle by replacing the optimization over $\mathcal{D}$ in Equation (5.2) with universal function approximators such as neural networks. Moreover, one can use existing global search techniques to implement the maximization oracle. While such a heuristic approach can reduce the computational complexity of the oracles, bounding their approximation errors can be hard (recall, the worst-case risk of our estimator depends on the approximation error of the optimization oracles). Nevertheless, in later sections, we empirically demonstrate that the estimators from this approach have superior performance over many existing estimators which are known to be approximately minimax.

We briefly discuss some classical work that can be leveraged for efficient implementation of optimization oracles, albeit for specific models or settings. For several problems, it can be shown that there exists an approximate minimax estimator in some restricted space of estimators such as linear or polynomial functions of the data [19, 29, 88]. Such results can be used to reduce the space of estimators in the statistical game (1.1). By replacing $\mathcal{M}_{\mathcal{D}}$ in Equation (1.1) with the restricted estimator space, one can greatly reduce the computational complexity of the optimization oracles. Another class of results relies on analyses of convergence of posterior distributions. As a key instance, when the number of samples n is much larger than the dimension d, it is well known that the posterior distribution behaves like a normal distribution, whenever the prior has sufficient mass around the true parameter [46]. Such a property can be used to efficiently implement the minimization oracle.

5.2 Invariance of Minimax Estimators and LFPs

In this section, we show that whenever the statistical game satisfies certain invariance properties, the computational complexity of the optimization oracles required by Algorithm 5.1 can be greatly reduced. We first present a classical result from statistics about the invariance properties of minimax estimators.When the statistical game in Equation (1.2) is invariant to group transformations, the *invariance theorem* says that there exist minimax estimators which are also invariant to these group transformations [9, 62]. Later, we utilize this result to reduce the computational complexity of the oracles required by Algorithm 5.1.

We first introduce the necessary notation and terminology to formally state the invariance theorem. We note that the theorem stated here is tailored for our setting and more general

versions of the theorem can be found in Kiefer et al. [62]. Let G be a compact group of transformations on $\mathcal{X} \times \Theta$ which acts component wise; that is, for each $g \in G$, $g(X, \theta)$ can be written as $(g_1 X, g_2 \theta)$, where g_1, g_2 are transformations on $\mathcal{X}, \Theta$. With a slight abuse of notation we write $gX, g\theta$ in place of $g_1 X, g_2 \theta$. We assume that the group action is continuous, so that the functions $(g, X) \to gX$ and $(g, \theta) \to g\theta$ are continuous. Finally, let μ be the unique left Haar measure on G with $\mu(G) = 1$. We now formally define "invariant statistical games", "invariant estimators" and "invariant probability measures".

Definition 5.2.1 (Invariant Game)**.** A statistical game is invariant to group transformations G, if the following two conditions hold for each $g \in G$

- for all $\theta \in \Theta$, $g\theta \in \Theta$. Moreover, the probability distribution of gX is $P_{g\theta}$, whenever the distribution of X is P_θ.
- $M(g\theta_1, g\theta_2) = M(\theta_1, \theta_2)$, for all $\theta_1, \theta_2 \in \Theta$.

Definition 5.2.2 (Invariant Estimator)**.** A deterministic estimator $\hat{\theta}$ is invariant if for each $g \in G$, $\hat{\theta}(g\mathbb{X}^n) = g\hat{\theta}(\mathbb{X}^n)$, where $g\mathbb{X}^n = \{gX_1, \dots gX_n\}$.

Definition 5.2.3 (Invariant Measure)**.** Let $\mathcal{B}(\Theta)$ be the Borel σ-algebra corresponding to the parameter space Θ. A measure ν on $(\Theta, \mathcal{B}(\Theta))$ is invariant if for all $g \in G$ and any measurable set $A \in \mathcal{B}(\Theta)$, $\nu(gA) = \nu(A)$.

Example 5.2.1. Consider the problem of estimating the mean of a Gaussian distribution. Given n samples $X_1, \dots X_n$ drawn from $\mathcal{N}(\theta, I_{d\times d})$, our goal is to estimate the unknown parameter θ. Suppose the parameter space is given by $\Theta = \{\theta' : \|\theta'\|_2 \leq B\}$ and the risk of any estimator is measured w.r.t squared L_2 loss. Then it is easy to verify that the problem is invariant to transformations of the orthogonal group $\mathbb{O}(d) = \{U : UU^T = U^T U = I\}$.

We now present the main result concerning the existence of invariant minimax estimators.A more general version of the result can be found in [62].

Theorem 5.2 (Invariance)**.** *Consider the statistical game in Equation (1.1). Suppose the game is invariant to group transformations G. Suppose the loss metric M is convex in its first argument. Then for any deterministic estimator $\hat{\theta}$, there exists an estimator $\hat{\theta}_G$ which is invariant to group transformations G, with worst-case risk no larger than the worst-case risk of $\hat{\theta}$*

$$\sup_{\theta\in\Theta} R(\hat{\theta}_G, \theta) \leq \sup_{\theta\in\Theta} R(\hat{\theta}, \theta).$$

This shows that there exists a minimax estimator which is invariant to group transformations. We now utilize this invariance property to reduce the complexity of the optimization oracles. Let $\Theta = \bigcup_\beta \Theta_\beta$ be the partitioning of Θ into equivalence classes under the equivalence $\theta_1 \sim \theta_2$, if $\theta_1 = g\theta_2$ for some $g \in G$. The quotient space of Θ is defined as the set of equivalence classes of the elements of Θ under the above defined equivalence and is given by $\Theta/G = \{\Theta_\beta\}_\beta$. For an invariant estimator $\hat{\theta}$, we define $R_G(\hat{\theta}, \Theta_\beta)$ as $R(\hat{\theta}, \theta_\beta)$ for any $\theta_\beta \in \Theta_\beta$. Note that this is well defined because for invariant estimators $R(\hat{\theta}, \theta_1) = R(\hat{\theta}, \theta_2)$ whenever $\theta_1 \sim \theta_2$ (see Lemma 5.1). Our main result shows that Equation (1.1) can be reduced to the following simpler objective

$$\inf_{\hat{\theta}\in\mathcal{M}_{\mathcal{D},G}} \sup_{\Theta_\beta\in\Theta/G} R_G(\hat{\theta}, \Theta_\beta), \tag{5.3}$$

where $\mathcal{M}_{\mathcal{D},G}$ represents the set of randomized estimators which are invariant to group transformations G. This shows that the outer minimization over the set of all estimators in Equation (1.1) can be replaced with a minimization over just the invariant estimators. Moreover, the inner maximization over the entire parameter space Θ can be replaced with a maximization over the smaller quotient space Θ/G , which in many examples we study here is a one or two-dimensional space, irrespective of the dimension of Θ.

Theorem 5.3. *Suppose the statistical game in Equation (1.1) is invariant to group transformations G. Moreover, suppose the loss metric M is convex in its first argument. Then,*

$$\inf_{\hat{\theta}\in\mathcal{M}_{\mathcal{D}}} \sup_{\theta\in\Theta} R(\hat{\theta},\theta) = \inf_{\hat{\theta}\in\mathcal{M}_{\mathcal{D},G}} \sup_{\Theta_\beta\in\Theta/G} R_G(\hat{\theta},\Theta_\beta).$$

Moreover, given any ϵ-approximate mixed strategy NE of the reduced statistical game (5.3), one can reconstruct an ϵ-approximate mixed strategy NE of the original statistical game (1.1).

We now demonstrate how Theorem 5.3 can be used on a variety of fundamental statistical estimation problems.

5.2.1 Finite Gaussian Sequence Model

In the finite Gaussian sequence model, we are given a single sample $X \in \mathbb{R}^d$ sampled from a Gaussian distribution $\mathcal{N}(\theta, I)$. We assume the parameter θ has a bounded L_2 norm and satisfies $\|\theta\|_2 \leq B$. Our goal is to design an estimator for θ which is minimax with respect to squared-error loss. This results in the following min-max problem

$$\inf_{\hat{\theta}\in\mathcal{M}_{\mathcal{D}}} \sup_{\|\theta\|_2\leq B} R(\hat{\theta},\theta) \equiv \mathbb{E}_{X\sim\mathcal{N}(\theta,I)}\left[\|\hat{\theta}(X)-\theta\|_2^2\right]. \tag{5.4}$$

Theorem 5.4. *Let $\mathbb{O}(d) = \{U : UU^T = U^TU = I\}$ be the group of $d\times d$ orthogonal matrices with matrix multiplication as the group operation. The statistical game in Equation (5.4) is invariant under the action of $\mathbb{O}(d)$, where the action of $g \in \mathbb{O}(d)$ on (X,θ) is defined as $g(X,\theta) = (gX, g\theta)$. Moreover, the quotient space $\Theta/\mathbb{O}(d)$ is homeomorphic to the real interval $[0,B]$ and the reduced statistical game is given by*

$$\inf_{\hat{\theta}\in\mathcal{M}_{\mathcal{D},G}} \sup_{b\in[0,B]} R(\hat{\theta}, b\mathbf{e}_1), \tag{5.5}$$

where $\mathbf{e}_1$ is the first standard basis vector in $\mathbb{R}^d$ and $\mathcal{M}_{\mathcal{D},G}$ represents the set of randomized estimators which are invariant to the actions of orthogonal group.

The theorem shows that the supremum in the reduced statistical game (5.3) is over a bounded interval on the real line. So the maximization oracle in this case can be efficiently implemented using grid search over the interval $[0,B]$. In Section 5.3 we use this result to obtain estimators for Gaussian sequence model which are provably minimax and can be computed in polynomial time.

Estimating a few co-ordinates. Here, we again consider with the Gaussian sequence model described above, but we are now interested in the estimation of only a subset of the co-ordinates of θ. Without loss of generality, we assume these are the first k coordinates. The loss M is the squared L_2 loss on the first k coordinates. The following Theorem presents the invariance properties of this problem. It relies on the group $\mathbb{O}(k) \times \mathbb{O}(d-k)$, which is defined as the set of orthogonal matrices of the form $g = \begin{bmatrix} g_1 & 0 \\ 0 & g_2 \end{bmatrix}$ where $g_1 \in \mathbb{O}(k)$ and $g_2 \in \mathbb{O}(d-k)$.

Theorem 5.5. *The statistical game described above is invariant under the action of the group $\mathbb{O}(k) \times \mathbb{O}(d-k)$. Moreover, the quotient space $\Theta/\mathbb{O}(k) \times \mathbb{O}(d-k)$ is homeomorphic to the ball of radius B centered at origin in $\mathbb{R}^2$ and the reduced statistical game is given by*

$$\inf_{\hat{\theta} \in \mathcal{M}_{\mathcal{D},G}} \sup_{b_1^2 + b_2^2 \leq B^2} R(\hat{\theta}, [b_1 \mathbf{e}_{1,k}, b_2 \mathbf{e}_{1,d-k}]), \tag{5.6}$$

where $\mathbf{e}_{1,k}$ is the first standard basis vector in $\mathbb{R}^k$ and $\mathcal{M}_{\mathcal{D},G}$ represents the set of randomized estimators which are invariant to the actions of orthogonal group.

5.2.2 Linear Regression

In the problem of linear regression with random design we are given n independent samples $D_n = \{(X_i, Y_i)\}_{i=1}^n$ generated from a linear model $Y_i = X_i^T \theta^* + \epsilon_i$, where $X_i \sim \mathcal{N}(0, I)$, and $\epsilon_i \sim \mathcal{N}(0, 1)$. We assume the true regression vector is bounded and satisfies $\|\theta^*\|_2 \leq B$. Our goal is to design minimax estimator for estimating θ^* from D_n, w.r.t squared error loss. This leads us to the following min-max problem

$$\inf_{\hat{\theta} \in \mathcal{M}_{\mathcal{D}}} \sup_{||\theta||_2 \leq B} R(\hat{\theta}, \theta) \equiv \mathbb{E}_{D_n} \left[||\hat{\theta}(D_n) - \theta||_2^2 \right]. \tag{5.7}$$

Theorem 5.6. *The statistical game in Equation (5.7) is invariant under the action of the orthogonal group $\mathbb{O}(d)$, where the action of $g \in \mathbb{O}(d)$ on $((X, Y), \theta)$ is defined as $g((X, Y), \theta) = ((gX, Y), g\theta)$. Moreover, the quotient space $\Theta/\mathbb{O}(d)$ is homeomorphic to the interval $[0, B]$ and the reduced statistical game is given by*

$$\inf_{\hat{\theta} \in \mathcal{M}_{\mathcal{D},G}} \sup_{b \in [0,B]} R(\hat{\theta}, b\mathbf{e}_1), \tag{5.8}$$

where $\mathcal{M}_{\mathcal{D},G}$ represents the set of randomized estimators which are invariant to the actions of orthogonal group.

5.2.3 Normal Covariance Estimation

In the problem of normal covariance estimation we are given n independent samples $\mathbb{X}^n = \{X_i\}_{i=1}^n$ drawn from $N(0, \Sigma)$. Here, we assume that the true Σ has a bounded operator norm and satisfies $\|\Sigma\|_2 \leq B$. Our goal is to construct an estimator for Σ which is minimax w.r.t the entropy loss, which is defined as

$$M(\Sigma_1, \Sigma_2) = \operatorname{tr}\left(\Sigma_1^{-1}\Sigma_2\right) - \log|\Sigma_1^{-1}\Sigma_2| - d.$$

This leads us to the following min-max problem

$$\inf_{\hat{\Sigma}\in\mathcal{M}_{\mathcal{D}}} \sup_{\Sigma\in\Xi} R(\hat{\Sigma},\Sigma) \equiv \mathbb{E}_{\mathbb{X}^n}\left[M(\hat{\Sigma}(\mathbb{X}^n),\Sigma)\right], \tag{5.9}$$

where $\Xi = \{\Sigma : ||\Sigma||_2 \leq B\}$.

Theorem 5.7. *The statistical game defined by normal covariance estimation with entropy loss is invariant under the action of the orthogonal group $\mathbb{O}(d)$, where the action of $g \in \mathbb{O}(d)$ on (X,Σ) is defined as $g(X_i,\Sigma) = (gX_i, g\Sigma g^T)$. Moreover the quotient space $\Xi/\mathbb{O}(d)$ is homeomorphic to $\Xi_G = \{\lambda \in \mathbb{R}^d : B \geq \lambda_1 \geq \ldots \lambda_d > 0\}$ and the reduced statistical game is given by*

$$\inf_{\hat{\Sigma}\in\mathcal{M}_{\mathcal{D},G}} \sup_{\lambda\in\Xi_G} R(\hat{\Sigma}, Diag(\lambda)), \tag{5.10}$$

where $Diag(\lambda)$ is the diagonal matrix whose diagonal entries are given by λ and $\mathcal{M}_{\mathcal{D},G}$ represents the set of randomized estimators which are invariant to the actions of orthogonal group.

The theorem shows that the maximization problem over Ξ can essentially be reduced to an optimization problem over a d-dimensional space.

5.2.4 Entropy estimation

In the problem of entropy estimation, we are given n samples $\mathbb{X}^n = \{X_1, \ldots X_n\}$ drawn from a discrete distribution $P = (p_1, \ldots p_d)$. Here, the domain of each X_i is given by $\mathcal{X} = \{1, 2, \ldots d\}$. Our goal is to estimate the entropy of P, which is defined as $f(P) = -\sum_{i=1}^{d} p_i \log_2 p_i$, under the squared error loss. This leads us to the following min-max problem

$$\inf_{\hat{f}\in\mathcal{M}_{\mathcal{D}}} \sup_{P\in\mathcal{P}} R(\hat{f}, P) \equiv \mathbb{E}_{\mathbb{X}^n}\left[\left(\hat{f}(\mathbb{X}^n) - f(P)\right)^2\right], \tag{5.11}$$

where $\mathcal{P}$ is the set of all probability distributions supported on d elements.

Theorem 5.8. *The statistical game in Equation (5.11) is invariant to the action of the permutation group $\mathbb{S}_d$. The quotient space $\mathcal{P}/\mathbb{S}_d$ is homeomorphic to $\mathcal{P}_G = \{P \in \mathbb{R}^d : 1 \geq p_1 \geq \ldots \geq p_d \geq 0,\ \sum_i p_i = 1\}$ and the reduced statistical game is given by*

$$\inf_{\hat{f}\in\mathcal{M}_{\mathcal{D},G}} \sup_{P\in\mathcal{P}_G} R(\hat{f}, P), \tag{5.12}$$

where $\mathcal{M}_{\mathcal{D},G}$ represents the set of randomized estimators which are invariant to the actions of permutation group.

5.3 Finite Gaussian Sequence Model

In this section we consider the finite Gaussian sequence model described in Section 5.2.1 and use Algorithm 5.1 to construct a provably minimax estimator, which can be computed in polynomial time. This problem has received a lot of attention in statistics because of its simplicity, relevance and its connections to non-parametric regression [see Chapter 1 of 56]. When the radius of the domain B is smaller than $1.15\sqrt{d}$, Marchand and Perron [78] show that the Bayes estimator with uniform prior on the boundary is a minimax estimator for the

problem. For larger values of B, the exact minimax estimator is unknown. Several works have attempted to understand the properties of LFP in such settings [20] and constructed approximate minimax estimators [12]. In this work, we rely on Algorithm 5.1 to construct an exact minimax estimator and an LFP, for any value of B, d.

Recall, in Theorem 5.4 we showed that the original min-max statistical game can be reduced to the simpler problem in Equation (5.5) To use Algorithm 5.1 to find a Nash equilibrium of the reduced game, we need efficient implementation of the required optimization oracles and a bound on their approximation errors. The optimization problems corresponding to the oracles in Equations (5.1), (5.2) are given as follows

$$\hat{\theta}_t \leftarrow \operatorname*{argmin}_{\hat{\theta} \in \mathcal{D}_G} \mathbb{E}_{b \sim P_t} \left[R(\hat{\theta}, b\mathbf{e}_1) \right], \quad b_t(\sigma) \leftarrow \operatorname*{argmax}_{b \in [0,B]} \sum_{i=1}^{t-1} R(\hat{\theta}_i, b\mathbf{e}_1) + \sigma b,$$

where $\mathcal{D}_G$ is the set of deterministic invariant estimators and P_t is the distribution of random variable $b_t(\sigma)$. We now present efficient techniques for implementing these oracles (Algorithms 5.2, 5.3). Since the maximization problem is a 1 dimensional optimization problem, grid search can be used to compute an approximate maximizer. The approximation error of the resulting oracle depends on the grid width and the number of samples used to compute the expectation in the risk $R(\hat{\theta}, b\mathbf{e}_1)$. Later, we show that $\text{poly}(d, B)$ grid points and samples suffice to have a small approximation error. The minimization problem, which requires finding an invariant estimator minimizing the integrated risk under any prior P_t, can also be efficiently implemented. As shown in Proposition 5.2 below, the minimizer has a closed-form expression which depends on P_t and modified Bessel functions. To compute an approximate minimizer of the problem, we approximate P_t with its samples and rely on the closed-form expression. The approximation error of this oracle depends on the number of samples used to approximate P_t. We again show that $\text{poly}(d, B)$ samples suffice to have a small approximation error.

Proposition 5.2. *The optimizer $\hat{\theta}_t$ of the minimization problem defined above has the following closed-form expression*

$$\hat{\theta}_t(X) = \left(\frac{\mathbb{E}_{b \sim P_t} \left[b^{3-d/2} e^{-b^2/2} I_{d/2}(b\|X\|_2) \right]}{\mathbb{E}_{b \sim P_t} \left[b^{2-d/2} e^{-b^2/2} I_{d/2-1}(b\|X\|_2) \right]} \right) \frac{X}{\|X\|_2},$$

where I_ν is the modified Bessel function of first kind of order ν.

We now show that using Algorithm 5.1 for solving objective (5.5) with Algorithms 5.2, 5.3 as optimization oracles, gives us a provably minimax estimator and an LFP for finite Gaussian sequence model.

Theorem 5.9. *Suppose Algorithm 5.1 is run for T iterations with Algorithms 5.2, 5.3 as the maximization and minimization oracles. Suppose the hyper-parameters of these algorithms are set as $\eta = \frac{1}{B(B+1)\sqrt{T}}$, $w = \frac{B}{T^{3/2}}$, $N_1 = \frac{T^3}{(B+1)^2}$, $N_2 = \frac{T^4}{(B+1)^2}$. Let $\hat{P}_t$ be the approximation of probability distribution P_t used in the t^{th} iteration of Algorithm 5.1. Moreover, let $\hat{\theta}_t$ be the output of Algorithm 5.3 in the t^{th} iteration of Algorithm 5.1.*

Algorithm 5.2 Maximization Oracle

Input: Estimators $\{\hat{\theta}_i\}_{i=1}^{t-1}$, perturbation σ, grid width w, number of samples for computation of expected risk $R(\hat{\theta}, \theta)$: N_1
Let $\{b_1, b_2 \dots b_{B/w}\}$ be uniformly spaced points on $[0, B]$
for $j = 1 \dots B/w$ **do**
    for $i = 1 \dots t-1$ **do**
        Generate N_1 independent samples $\{X_k\}_{k=1}^{N_1}$ from the distribution $\mathcal{N}(b_j \mathbf{e}_1, I)$
        Estimate $R(\hat{\theta}_i, b_j \mathbf{e}_1)$ as $\frac{1}{N_1} \sum_{k=1}^{N_1} \|\hat{\theta}_i(X_k) - b\mathbf{e}_1\|_2^2$.
    Evaluate the objective at b_j using the above estimates
Output: b_j which maximizes the objective

Algorithm 5.3 Minimization Oracle

Input: Samples $\{b_i\}_{i=1}^{N_2}$ generated from distribution P_t.
For any X, compute $\hat{\theta}_t(X)$ as

$$\left(\frac{\sum_{i=1}^{N_2} w_i b_i A(b_i \|X\|_2)}{\sum_{i=1}^{N_2} w_i} \right) \frac{X}{\|X\|_2},$$

where $A(\gamma) = \dfrac{I_{d/2}(\gamma)}{I_{d/2-1}(\gamma)}$, $w_i = b_i^{2-d/2} e^{-b_i^2/2} I_{d/2-1}(b_i \|X\|_2)$, and I_ν is the modified Bessel function of the first kind of order ν.

1. *Then the averaged estimator* $\hat{\theta}_{avg}(X) = \frac{1}{T} \sum_{i=1}^{T} \hat{\theta}_i(X)$ *is approximately minimax and satisfies the following worst-case risk bound with probability at least* $1 - \delta$

$$\sup_{\theta : \|\theta\|_2 \le B} R(\hat{\theta}_{avg}, \theta) \le R^* + \tilde{O}\left(\frac{B^2(B+1)}{\sqrt{T}} \right),$$

   where $\tilde{O}(.)$ *hides log factors and* R^* *is the minimax risk.*

2. *Define the mixture distribution* $\hat{P}_{AVG}$ *as* $\frac{1}{T} \sum_{i=1}^{T} \hat{P}_i$. *Let* $\hat{P}_{LFP}$ *be a probability distribution over* $\mathbb{R}^d$ *with density function defined as* $\hat{p}_{LFP}(\theta) \propto \|\theta\|_2^{1-d} \hat{P}_{AVG}(\|\theta\|_2)$, *where* $\hat{P}_{AVG}(\|\theta\|_2)$ *is the probability mass placed by* $\hat{P}_{AVG}$ *at* $\|\theta\|_2$. *Then* $\hat{P}_{LFP}$ *is approximately least favorable and satisfies the following with probability at least* $1 - \delta$

$$\inf_{\hat{\theta} \in \mathcal{D}} R(\hat{\theta}, \hat{P}_{LFP}) \ge R^* - \tilde{O}\left(\frac{B^2(B+1)}{\sqrt{T}} \right),$$

   where the infimum is over the set of all estimators.

We believe the polynomial factors in the bounds can be improved with a tighter analysis of the algorithm. The above Theorem shows that Algorithm 5.1 learns an approximate minimax estimator in poly(d, B) time. To the best our knowledge, this is the first result providing provable minimax estimators for finite Gaussian sequence model, for any value of B.

5.4 Linear Regression

In this section we consider the linear regression problem described in Section 5.2.2 and provide a provably minimax estimator. Recall, in Theorem 5.6 we showed that the original min-max statistical game can be reduced to the simpler problem in Equation (5.8). We now provide efficient implementations of the optimization oracles required by Algorithm 5.1 for finding a Nash equilibrium of this game. The optimization problems corresponding to the two optimization oracles are as follows

$$\hat{\theta}_t \leftarrow \operatorname*{argmin}_{\hat{\theta} \in \mathcal{D}_G} \mathbb{E}_{b \sim P_t} \left[R(\hat{\theta}, b\mathbf{e}_1) \right], \quad b_t(\sigma) \leftarrow \operatorname*{argmax}_{b \in [0,B]} \sum_{i=1}^{t-1} R(\hat{\theta}_i, b\mathbf{e}_1) + \sigma b,$$

where $\mathcal{D}_G$ is the set of deterministic invariant estimators and P_t is the distribution of random variable $b_t(\sigma)$. Similar to the Gaussian sequence model, the maximization oracle can be efficiently implemented via a grid search over $[0, B]$ (Algorithm 5.4). The solution to the minimization problem has a closed-form expression in terms of the mean and normalization constant of Fisher-Bingham distribution, which is a distribution obtained by constraining multivariate normal distributions to lie on the surface of unit sphere [68]. Letting $\mathbb{S}^{d-1}$ be the unit sphere in $\mathbb{R}^d$, the probability density of a random variable Z distributed according to Fisher-Bingham distribution is given by

$$p(Z; A, \gamma) = C(A, \gamma)^{-1} \exp\left(-Z^T A Z + \langle \gamma, Z \rangle\right),$$

where $Z \in \mathbb{S}^{d-1}$, and $\gamma \in \mathbb{R}^d$, $A \in \mathbb{R}^{d \times d}$ are the parameters of the distribution with A being positive semi-definite and $C(A, \gamma)$ is the normalization constant. Note that the mean of Fisher-Bingham distribution is given by $C(A, \gamma)^{-1} \frac{\partial}{\partial \gamma} C(A, \gamma)$. The following proposition obtains a closed-form expression for $\hat{\theta}_t$ in terms of $C(A, \gamma)$ and $\frac{\partial}{\partial \gamma} C(A, \gamma)$.

Proposition 5.3. *The optimizer $\hat{\theta}_t$ of the minimization problem defined above has the following closed-form expression*

$$\hat{\theta}_t(D_n) = \frac{\mathbb{E}_{b \sim P_t} \left[b^2 \frac{\partial}{\partial \gamma} C\left(2^{-1} b^2 \mathbf{X}^T \mathbf{X}, \gamma\right) \Big|_{\gamma = b\mathbf{X}^T \mathbf{Y}} \right]}{\mathbb{E}_{b \sim P_t} \left[b C\left(2^{-1} b^2 \mathbf{X}^T \mathbf{X}, b\mathbf{X}^T \mathbf{Y}\right) \right]},$$

where $\mathbf{X} = [X_1, X_2 \dots X_n]^T$ and $\mathbf{Y} = [Y_1, Y_2 \dots Y_n]$.

We note that there exist a number of efficient techniques for computation of the mean and normalization constant of Fisher-Bingham distribution [51, 68]. In our experiments we rely on the technique of Kume and Wood [68] (we relegate the details of this technique to Section 5.8.9.2). To compute an approximate optimizer of the minimization problem, we approximate P_t with its samples and rely on the above closed-form expression. Algorithm 5.5 describes the resulting minimization oracle. We now show that using Algorithm 5.1 for solving objective (5.8) with Algorithms 5.4, 5.5 as optimization oracles, gives us a provably minimax estimator and an LFP for linear regression.

Theorem 5.10. *Suppose Algorithm 5.1 is run for T iterations with Algorithms 5.4, 5.5 as the maximization and minimization oracles. Suppose the hyper-parameters of these algorithms are set as $\eta = \frac{1}{B(B\sqrt{n}+1)\sqrt{T}}$, $w = \frac{B}{T^{3/2}}$, $N_1 = \frac{T^3}{(B\sqrt{n}+1)^2}, N_2 = \frac{T^4}{(B\sqrt{n}+1)^2}$. Let $\hat{P}_t$ be*

Algorithm 5.4 Regression Maximization Oracle

Input: Estimators $\{\hat{\theta}_i\}_{i=1}^{t-1}$, perturbation σ, grid width w, number of samples for computation of expected risk $R(\hat{\theta}, \theta)$: N_1
Let $\{b_1, b_2 \dots b_{B/w}\}$ be uniformly spaced points on $[0, B]$
for $j = 1 \dots B/w$ **do**
    for $i = 1 \dots t-1$ **do**
        Generate N_1 independent datasets $\{D_{n,k}\}_{k=1}^{N_1}$ from the linear model with true regression vector $b_j \mathbf{e}_1$
        Estimate $R(\hat{\theta}_i, b_j \mathbf{e}_1)$ as $\frac{1}{N_1} \sum_{k=1}^{N_1} \|\hat{\theta}_i(D_{n,k}) - b\mathbf{e}_1\|_2^2$.
    Evaluate the objective at b_j using the above estimates
Output: b_j which maximizes the objective

Algorithm 5.5 Regression Minimization Oracle

Input: Samples $\{b_i\}_{i=1}^{N_2}$ generated from distribution P_t
For any D_n, compute $\hat{\theta}_t(D_n)$ as

$$\hat{\theta}_t(D_n) = \frac{\sum_{i=1}^{N_2} b_i^2 \frac{\partial}{\partial \gamma} C\left(2^{-1} b_i^2 \mathbf{X}^T \mathbf{X}, \gamma\right)\Big|_{\gamma = b_i \mathbf{X}^T \mathbf{Y}}}{\sum_{i=1}^{N_2} b_i C\left(2^{-1} b_i^2 \mathbf{X}^T \mathbf{X}, b_i \mathbf{X}^T \mathbf{Y}\right)},$$

where $\mathbf{X} = [X_1, X_2 \dots X_n]^T$ and $\mathbf{Y} = [Y_1, Y_2 \dots Y_n]$.

the approximation of probability distribution P_t used in the t^{th} iteration of Algorithm 5.1. Moreover, let $\hat{\theta}_t$ be the output of Algorithm 5.5 in the t^{th} iteration of Algorithm 5.1.

1. *Then the averaged estimator $\hat{\theta}_{avg}(D_n) = \frac{1}{T} \sum_{i=1}^{T} \hat{\theta}_i(D_n)$ is approximately minimax and satisfies the following worst-case risk bound with probability at least $1 - \delta$*

$$\sup_{\theta : \|\theta\|_2 \le B} R(\hat{\theta}_{avg}, \theta) \le R^* + \tilde{O}\left(B^2 (B+1) \sqrt{\frac{n}{T}}\right).$$

2. *Define the mixture distribution $\hat{P}_{AVG}$ as $\frac{1}{T} \sum_{i=1}^{T} \hat{P}_i$. Let $\hat{P}_{LFP}$ be a probability distribution over $\mathbb{R}^d$ with density function defined as $\hat{p}_{LFP}(\theta) \propto \|\theta\|_2^{1-d} \hat{P}_{AVG}(\|\theta\|_2)$, where $\hat{P}_{AVG}(\|\theta\|_2)$ is the probability mass placed by $\hat{P}_{AVG}$ at $\|\theta\|_2$. Then $\hat{P}_{LFP}$ is approximately least favorable and satisfies the following with probability at least $1 - \delta$*

$$\inf_{\hat{\theta} \in \mathcal{D}} R(\hat{\theta}, \hat{P}_{LFP}) \ge R^* - \tilde{O}\left(B^2 (B+1) \sqrt{\frac{n}{T}}\right).$$

5.5 Covariance Estimation

In this section, we consider the problem of normal covariance estimation. Recall, in Section 5.2.3 we showed that the problem is invariant to the action of the orthogonal group and can be reduced to the simpler problem in Equation (5.10). The optimization problems

corresponding to the oracles in Equations (5.1), (5.2) are as follows

$$\hat{\Sigma}_t \leftarrow \underset{\hat{\Sigma}\in\mathcal{D}_G}{\operatorname{argmin}}\, \mathbb{E}_{\lambda\sim P_t}\left[R(\hat{\Sigma}, \operatorname{Diag}(\lambda))\right], \quad \lambda_t(\sigma) \leftarrow \underset{\lambda\in\Xi_G}{\operatorname{argmax}} \sum_{i=1}^{t-1} R(\hat{\Sigma}_i, \operatorname{Diag}(\lambda)) + \langle\lambda, \sigma\rangle,$$

where $\mathcal{D}_G$ is the set of deterministic invariant estimators and P_t is the distribution of random variable $\lambda_t(\sigma)$. Note that the maximization problem involves optimization of a non-concave objective in d-dimensional space. So, implementing a maximization oracle with low approximation error can be computationally expensive, especially in high dimensions. Moreover, unlike finite Gaussian sequence model and linear regression, the minimization problem doesn't have a closed form expression, and it is not immediately clear how to efficiently implement a minimization oracle with low approximation error. In such scenarios, we show that one can rely on a combination of heuristics and problem structure to further reduce the computational complexity of the optimization oracles. Although relying on heuristics comes at the expense of theoretical guarantees, in later sections, we empirically demonstrate that the resulting estimators have superior performance over classical estimators. We begin by showing that the domain of the outer minimization in Equation (5.10) can be reduced to a smaller set of estimators. Our reduction relies on Blackwell's theorem, which shows that for convex loss functions M, there exists a minimax estimator which is a function of the sufficient statistic [50]. We note that Blackwell's theorem is very general and can be applied to a wide range of problems, to reduce the computational complexity of the minimization oracle.

Proposition 5.4. *Consider the problem of normal covariance estimation. Let* $S_n = \frac{\sum_{i=1}^n X_i X_i^T}{n}$ *be the empirical covariance matrix and let* $U\Delta U^T$ *be the eigen decomposition of* S_n*. Then there exists a minimax estimator which can be approximated arbitrarily well using estimators of the form* $\hat{\Sigma}_{f,g}(\mathbb{X}^n) = U\tilde{\Sigma}_{f,g}(\Delta)U^T$*, where* $\tilde{\Sigma}_{f,g}(\Delta)$ *is a diagonal matrix whose* i^{th} *diagonal entry is given by*

$$\tilde{\Sigma}_{f,g,i}(\Delta) = f\left(\Delta_i, \sum_{j\neq i} g(\Delta_i, \Delta_j)\right),$$

for some functions $f : \mathbb{R}^{d+1} \to \mathbb{R}, g : \mathbb{R}^2 \to \mathbb{R}^d$*. Here,* Δ_i *is the* i^{th} *diagonal entry of* Δ*. Moreover, the optimization problem in Equation (5.10) can be reduced to the following simpler problem*

$$\inf_{\hat{\Sigma}\in\mathcal{M}_{f,g}} \sup_{\lambda\in\Xi_G} R(\hat{\Sigma}, Diag(\lambda)) = R^*, \tag{5.13}$$

where $\mathcal{M}_{f,g}$ *is the set of probability distributions over estimators of the form* $\hat{\Sigma}_{f,g}$*.*

We now use Algorithm 5.1 to solve the statistical game in Equation (5.13). The optimization problems corresponding to the two optimization oracles are given by

$$\hat{f}_t, \hat{g}_t \leftarrow \underset{f,g}{\operatorname{argmin}}\, \mathbb{E}_{\lambda\sim P_t}\left[R(\hat{\Sigma}_{f,g}, \operatorname{Diag}(\lambda))\right], \quad \lambda_t(\sigma) \leftarrow \underset{\lambda\in\Xi_G}{\operatorname{argmax}} \sum_{i=1}^{t-1} R(\hat{\Sigma}_{\hat{f}_i,\hat{g}_i}, \operatorname{Diag}(\lambda)) + \langle\lambda, \sigma\rangle.$$

We rely on heuristics to efficiently implement these oracles. To implement the minimization oracle, we use neural networks (which are universal function approximators) to parameterize

functions f, g. Implementing the minimization oracle then boils down to the finding the parameters of these networks which minimize the objective. To implement the maximization oracle, we rely on global search techniques. In our experiments, we use DragonFly [58], which is a zeroth order optimization technique, to implement this oracle. Note that these heuristics do not come with any guarantees and as a result the oracles are not guaranteed to have a small approximation error. Despite this, we empirically demonstrate that the estimators learned using this approach have good performance.

5.6 Entropy Estimation

In this section, we consider the problem of entropy estimation. Recall, in Section 5.2.4 we showed that the problem is invariant to the action of permutation group and can be reduced to the simpler problem in Equation (5.12). Similar to the problem of covariance estimation, implementing the optimization oracles for this problem, with low approximation error, can be computationally expensive. So we again rely on heuristics and problem structure to reduce the computational complexity of optimization oracles.

Proposition 5.5. *Consider the problem of entropy estimation. Let $\hat{P}_n = (\hat{p}_1, \dots \hat{p}_d)$ be the observed empirical probabilities. Then there exists a minimax estimator which can be approximated arbitrarily well using estimators of the form $\hat{f}_{g,h}(\hat{P}_n) = g(\sum_{i=1}^{d} h(\hat{p}_i))$, for some functions $g : \mathbb{R}^{d+1} \to \mathbb{R}, h : \mathbb{R} \to \mathbb{R}^{d+1}$. Moreover, the optimization problem in Equation (5.12) can be reduced to the following problem*

$$\inf_{\hat{f} \in \mathcal{M}_{g,h}} \sup_{P \in \mathcal{P}_G} R(\hat{f}, P) = R^*, \tag{5.14}$$

where $\mathcal{M}_{g,h}$ is the set of probability distributions over estimators of the form $\hat{f}_{g,h}$.

The proof of this proposition is presented in Section 5.8.11.1. We now use Algorithm 5.1 to solve the statistical game in Equation (5.14). The optimization problems corresponding to the two optimization oracles are given by

$$\hat{g}_t, \hat{h}_t \leftarrow \operatorname*{argmin}_{g,h} \mathbb{E}_{P \sim P_t} \left[R(\hat{f}_{g,h}, P) \right], \quad P_t(\sigma) \leftarrow \operatorname*{argmax}_{P \in \mathcal{P}_G} \sum_{i=1}^{t-1} R(\hat{f}_{\hat{g}_i, \hat{h}_i}, P) + \langle P, \sigma \rangle ,$$

where P_t is the distribution of random variable $P_t(\sigma)$. To implement the minimization oracle, we use neural networks to parameterize functions g, h. To implement the maximization oracle, we rely on DragonFly.

5.7 Experiments

In this section, we present experiments showing performance of the proposed technique for constructing minimax estimators. While our primary focus is on the finite Gaussian sequence model and linear regression for which we provided provably minimax estimators, we also present experiments on other problems such as covariance and entropy estimation. For each of these problems, we begin by describing the setup as well as the baseline algorithms, before proceeding to a discussion of the experimental findings.

5.7.1 Finite Gaussian Sequence Model

In this section, we focus on experiments related to the finite Gaussian sequence model. We first consider the case where the risk is measured with respect to squared error loss, *i.e.,* $M(\theta_1, \theta_2) = \|\theta_1 - \theta_2\|_2^2$.

Proposed Technique. We use Algorithm 5.1 with optimization oracles described in Algorithms 5.2, 5.3 to find minimax estimators for this problem. We set the hyper-parameters of our algorithm as follows: number of iterations of FTPL $T = 500$, grid width $w = 0.05 \times B$, number of samples for computation of $R(\hat{\theta}, \theta)$ in Algorithm 5.2 $N_1 = 1000$, number of samples generated from P_t in Algorithm 5.3 $N_2 = 1000$. We note that these are default values and were not tuned. The randomness parameter η in Algorithm 5.1 was tuned using a coarse grid search. We report the performance of the following two estimators constructed using the iterates of Algorithm 5.1: (a) Averaged Estimator $\hat{\theta}_{AVG}(X) = \frac{1}{T}\sum_{i=1}^{T} \hat{\theta}_i(X)$, (b) Bayes estimator for prior $\frac{1}{T}\sum_{i=1}^{T} \hat{P}_i$ which we refer to as "Bayes estimator for avg. prior". The performance of the randomized estimator $\hat{\theta}_{RND}$ is almost identical to the performance of $\hat{\theta}_{AVG}$. So we do not report its performance.

Baselines. We compare our estimators with various baselines: (a) standard estimator $\hat{\theta}(X) = X$, (b) James Stein estimator $\hat{\theta}(X) = \left(1 - (d-3)/\|X\|_2^2\right)^+ X$, where $c^+ = \max(0, c)$, (c) projection estimator (MLE) $\hat{\theta}(X) = \min(\|X\|_2, B)\frac{X}{\|X\|_2}$, (d) Bayes estimator for uniform prior on the boundary; this estimator is known to be minimax for $B \leq 1.15\sqrt{d}$.

Worst-case Risk. We compare the performance of various estimators based on their worst-case risk. The worst-case risk of the standard estimator is equal to d. The worst case risk of all the other estimators is computed as follows. Since all these estimators are invariant to orthogonal group transformations, the risk $R(\hat{\theta}, \theta)$ only depends on $\|\theta\|_2$ and not its direction. So the worst-case risk can be obtained by solving the following optimization problem: $\max_{b \in [0,B]} R(\hat{\theta}, b\mathbf{e}_1)$, where $\mathbf{e}_1$ is the first standard basis vector. We use grid search to solve this problem, with $0.05 \times B$ grid width. We use 10^4 samples to approximately compute $R(\hat{\theta}, b\mathbf{e}_1)$ for any $\hat{\theta}, b$.

Duality Gap. For estimators derived from our technique, we also present the duality gap, which is defined as $\sup_{\theta \in \Theta} R(\hat{\theta}_{AVG}, \theta) - \inf_{\hat{\theta} \in \mathcal{D}} R(\hat{\theta}, \frac{1}{T}\sum_{i=1}^{T} \hat{P}_i)$. Duality gap quantifies the closeness of $(\hat{\theta}_{AVG}, \frac{1}{T}\sum_{i=1}^{T} \hat{P}_i)$ to a Nash equilibrium. Smaller the gap, closer we are to an equilibrium.

Results. Table 5.1 shows the performance of various estimators for various values of d, B along with the duality gap for our estimator. For $B = \sqrt{d}$, the estimators obtained using Algorithm 5.1 have similar performance as the "Bayes estimator for uniform prior on boundary", which is known to be minimax. For $B = 2\sqrt{d}, 3\sqrt{d}$ for which the exact minimax estimator is unknown, we achieve better performance than baselines. Finally, we note that the duality gap numbers presented in the table can be made smaller by running our algorithm for more iterations. When the dimension $d = 1$, Donoho et al. [29] derived lower bounds for the minimax risk, for various values of B. In Table 5.2, we compare the worst risk of our estimator with these established lower bounds. It can be seen that the worst case risk of our estimator is close to the lower bounds.

Table 5.1: Worst-case risk of various estimators for finite Gaussian sequence model. The risk is measured with respect to squared error loss. The worst-case risk of the estimators from Algorithm 5.1 (last two rows) is smaller than the worst-case risk of baselines. The numbers in the brackets for Averaged Estimator represent the duality gap.

	Worst-case Risk								
	$B = \sqrt{d}$			**$B = 1.5\sqrt{d}$**			**$B = 2\sqrt{d}$**		
Estimator	**d = 10**	**d = 20**	**d = 30**	**d = 10**	**d = 20**	**d = 30**	**d = 10**	**d = 20**	**d = 30**
Standard	10	20	30	10	20	30	10	20	30
James Stein	6.0954	11.2427	16.073	7.9255	15.0530	21.3410	8.7317	16.6971	24.7261
Projection	8.3076	17.4788	26.7873	10.3308	20.3784	30.2464	10.1656	20.2360	30.3805
Bayes estimator for uniform prior on boundary	**4.8559**	**9.9909**	**14.8690**	11.7509	23.4726	35.2481	24.5361	49.0651	73.3158
Averaged Estimator	**4.7510** (0.1821)	**9.7299** (0.2973)	**14.8790** (0.0935)	**6.7990** (0.0733)	**13.8084** (0.2442)	**20.5704** (0.0087)	**7.8504** (0.3046)	**15.6686** (0.2878)	**23.8758** (0.6820)
Bayes estimator for avg. prior	**4.9763**	**10.1273**	**14.8128**	**6.7866**	**13.8200**	**20.3043**	**7.8772**	**15.6333**	**23.5954**

Table 5.2: Comparison of the worst case risk of $\hat{\theta}_{AVG}$ with established lower bounds from [29] for finite Gaussian sequence model with $d = 1$.

	B = 1	**B = 2**	**B = 3**	**B = 4**
Worst case risk of Averaged Estimator	0.456	0.688	0.799	0.869
Lower bound	0.449	0.644	0.750	0.814

5.7.2 Finite Gaussian Sequence Model with a few coordinates

In this section we again consider the finite Gaussian sequence model, but with a different risk. We now measure the risk on only the first k coordinates: $M(\theta_1, \theta_2) = \sum_{i=1}^{k}(\theta_1(i) - \theta_2(i))^2$. We present experimental results for $k = 1, d/2$.

Proposed Technique. Following Theorem 5.5, the original min-max objective can be reduced to the simpler problem in Equation (5.6). We use similar optimization oracles as in Algorithms 5.2, 5.3, to solve this problem. The maximization problem is now a 2D optimization problem for which we use grid search. The minimization problem, which requires computation of Bayes estimators, can be solved analytically and has similar expression as the Bayes estimator in Algorithm 5.3 (see Section 5.8.8 for details). We use a 2D grid of $0.05B$ width and length in the maximization oracle. We use the same hyper-parameters as above and run FTPL for 10000 iterations for $k = 1$ and 4000 iterations for $k = d/2$.

Worst-case Risk. We compare our estimators with the same baselines described in the previous section. For the case of $k = 1$, we also compare with the best linear estimator, which is known to be approximately minimax with worst case risk smaller than 1.25 times the minimax risk [28]. Since all these estimators, except the best linear estimator, are invariant to the transformations of group $\mathbb{O}(k) \times \mathbb{O}(d-k)$, the max risk of these estimators can be written as $\max_{b_1^2+b_2^2 \le B^2} R(\hat{\theta}, [b_1\mathbf{e}_{1,k}, b_2\mathbf{e}_{1,d-k}])$. We solve this problem using $2D$ grid search. The worst case risk of best linear estimator has a closed form expression.

Results. Table 5.3 shows the performance of various estimators for various values of d, B. It can be seen that for $B = \sqrt{d}$, our estimators have better performance than other baselines. The performance difference goes down for large B, which is as expected. In

Table 5.3: Worst-case risk of various estimators for bounded normal mean estimation when the risk is evaluated with respect to squared loss on the first k coordinates.

	Worst-case Risk								
	$\mathbf{k=1, B=\sqrt{d}}$			$\mathbf{k=1, B=2\sqrt{d}}$			$\mathbf{k=1, B=3\sqrt{d}}$		
Estimator	$\mathbf{d=10}$	$\mathbf{d=20}$	$\mathbf{d=30}$	$\mathbf{d=10}$	$\mathbf{d=20}$	$\mathbf{d=30}$	$\mathbf{d=10}$	$\mathbf{d=20}$	$\mathbf{d=30}$
Standard Estimator	1	1	1	1	1	1	1	1	1
James-Stein Estimator	2.3796	4.9005	7.3489	2.5087	4.9375	7.3760	2.4288	4.8951	7.3847
Projection Estimator	1.0055	1.4430	2.0424	1.0263	1.1051	1.5077	1.0288	1.0310	1.0202
Best Linear Estimator	0.9091	0.9524	0.9677	0.9756	0.9877	0.9917	0.9890	0.9945	0.9963
Bayes Estimator for average prior	**0.7955**	**0.8565**	**0.8996**	**0.9160**	**0.9496**	0.9726	0.9611	1.0007	1.0172
Averaged Estimator	**0.7939**	**0.8579**	**0.8955**	**0.9104**	**0.9497**	0.9724	0.9640	1.0003	1.0101

	Worst-case Risk								
	$\mathbf{k=d/2, B=\sqrt{d}}$			$\mathbf{k=d/2, B=2\sqrt{d}}$			$\mathbf{k=d/2, B=3\sqrt{d}}$		
Estimator	$\mathbf{d=10}$	$\mathbf{d=20}$	$\mathbf{d=30}$	$\mathbf{d=10}$	$\mathbf{d=20}$	$\mathbf{d=30}$	$\mathbf{d=10}$	$\mathbf{d=20}$	$\mathbf{d=30}$
Standard Estimator	5	10	15	5	10	15	5	10	15
James-Stein Estimator	4.1167	7.9200	11.6892	5.0109	9.7551	14.6568	5.0281	10.0155	14.9390
Projection Estimator	7.1096	15.8166	24.8158	30.3166	66.1806	103.0456	73.4834	156.5076	241.1031
Bayes Estimator for average prior	**3.2611**	**6.5834**	**9.8189**	**4.2477**	**8.6564**	**13.0606**	**4.6359**	**9.2773**	**13.9678**
Averaged Estimator	**3.2008**	**6.4763**	**9.7763**	**4.2260**	**8.6421**	**13.0353**	**4.6413**	**9.2760**	**13.9446**

order to gain insights about the estimator learned by our algorithm, we plot the contours of $\hat{\theta}_{AVG}(X)$ in Figure 5.1, for the $k = 1$ case, where the risk is measured on the first coordinate. It can be seen that when $X(1)$ is close to 0, irrespective of other coordinates, the estimator just outputs $X(1)$ as its estimate of $\theta(1)$. When $X(1)$ if far from 0, by looking along the corresponding vertical line, the estimator can be seen as outputting a shrinked version of $X(1)$, where the amount of shrinkage increases with the norm of $X(2:d)$. Note that this is unlike James Stein estimator which shrinks vectors with smaller norm more than larger norm vectors.

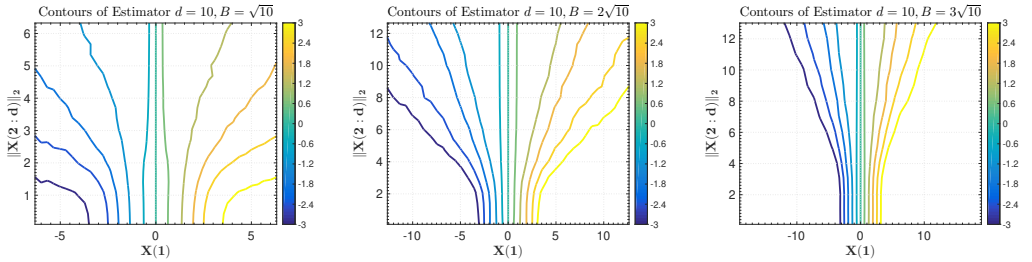


Figure 5.1: Contour plots of the estimator learned using Algorithm 5.1 when the risk is evaluated on the first coordinate. x axis shows the first coordinate of X, which is the input to the estimator. y axis shows the norm of the rest of the coordinates of X. The contour bar shows $\hat{\theta}(1)$, the first co-ordinate of the output of the estimator.

5.7.3 Linear Regression

In this section we present experimental results on linear regression. We use Algorithm 5.1 with optimization oracles described in Algorithms 5.4, 5.5 to find minimax estimators for this problem. We use the same hyper-parameter settings as finite Gaussian sequence model, and run Algorithm 5.1 for $T = 500$ iterations. We compare the worst-case risk of minimax estimators obtained using our algorithm for various values of (n, d, B), with ordinary least squares (OLS) and ridge regression estimators. Since all the estimators are

invariant to the transformations of orthogonal group $\mathbb{O}(d)$, the max risk can be written as $\max_{b\in[0,B]} R(\hat{\theta}, b\mathbf{e}_1)$, which can be efficiently computed using grid search. Table 5.4 presents the results from this experiment. It can be seen that we achieve better performance than ridge regression for small values of n/d, B. For large values of n/d, B, the performance of our estimator approaches ridge regression. The duality gap numbers presented in the Table suggest that the performance of our estimator can be improved for larger values of $n/d, B$, by choosing better hyper-parameters.

Table 5.4: Worst-case risk of various estimators for linear regression. The performance of ridge is obtained by choosing the best regularization parameter. The numbers in the brackets for Averaged Estimator represent the duality gap.

	Worst-case Risk							
	$\mathbf{n = 1.5 \times d, B = 0.5 \times \sqrt{d}}$				$\mathbf{n = 1.5 \times d, B = \sqrt{d}}$			
Estimator	$\mathbf{d = 5}$	$\mathbf{d = 10}$	$\mathbf{d = 15}$	$\mathbf{d = 20}$	$\mathbf{d = 5}$	$\mathbf{d = 10}$	$\mathbf{d = 15}$	$\mathbf{d = 20}$
OLS	5.0000	2.5000	2.5000	2.2222	5.0000	2.5000	2.5000	2.2222
Ridge regression	0.6637	0.9048	1.1288	1.1926	1.3021	1.4837	1.6912	1.6704
Averaged Estimator	**0.5827** (0.0003)	**0.8275** (0.0052)	**0.9839** (0.0187)	**1.0946** (0.0404)	**1.2030** (0.0981)	1.4615 (0.1145)	1.6178 (0.1768)	1.6593 (0.1863)
Bayes estimator for avg. prior	**0.5827**	**0.8275**	**0.9844**	**1.0961**	**1.1750**	1.4621	1.6265	1.6674

	Worst-case Risk							
	$\mathbf{n = 2 \times d, B = 0.5 \times \sqrt{d}}$				$\mathbf{n = 2 \times d, B = \sqrt{d}}$			
Estimator	$\mathbf{d = 5}$	$\mathbf{d = 10}$	$\mathbf{d = 15}$	$\mathbf{d = 20}$	$\mathbf{d = 5}$	$\mathbf{d = 10}$	$\mathbf{d = 15}$	$\mathbf{d = 20}$
OLS	1.2500	1.1111	1.0714	1.053	1.2500	1.1111	1.0714	1.053
Ridge regression	0.5225	0.6683	0.7594	0.8080	0.8166	0.8917	0.9305	0.9608
Averaged Estimator	0.4920 (0.0038)	**0.5991** (0.0309)	**0.6873** (0.0485)	**0.7339** (0.0428)	0.8044 (0.0647)	0.8615 (0.0854)	0.9388 (0.0996)	0.9621 (0.1224)
Bayes estimator for avg. prior	0.4894	**0.6004**	**0.6879**	**0.7320**	0.8140	0.8618	0.9375	0.9656

5.7.4 Covariance Estimation

In this section we present experimental results on normal covariance estimation.

Minimization oracle. In our experiments we use neural networks, which are universal function approximators, to parameterize functions f, g in Equation (5.13). To be precise, we use two layer neural networks to parameterize each of these functions. Implementing the minimization oracle then boils down to finding the parameters of these networks which minimize $\mathbb{E}_{\lambda\sim P_t}\left[R(\hat{\Sigma}_{f,g}, \mathrm{Diag}(\lambda))\right]$. In our experiments, we use stochastic gradient descent to learn these parameters.

Baselines. We compare the performance of the estimators returned by Algorithm 5.1 for various values of (n, d, B), with empirical covariance S_n and the James Stein estimator [52] which is defined as $K_n \Delta_{JS} K_n^T$, where K_n is a lower triangular matrix such that $S_n = K_n K_n^T$ and Δ_{JS} is a diagonal matrix with i^{th} diagonal element equal to $\frac{1}{n+d-2i+1}$.

Results. We use worst-case risk to compare the performance of various estimators. To compute the worst-case risk, we again rely on DragonFly. We note that the worst-case computed using this approach may be inaccurate as DragonFly is not guaranteed to return a global optimum. So, we also compare the risk of various estimators at randomly generated Σ's (see Section 5.8.12). Table 5.5 presents the results from this experiment. It can be seen that our estimators outperform empirical covariance for almost all the values of n, d, B and outperform James Stein estimator for small values of n/d, B. For large values of n/d, B,

our estimator has similar performance as JS. In this setting, we believe the performance of our estimators can be improved by running the algorithm with better hyper-parameters.

Table 5.5: Worst-case risk of various estimators for covariance estimation for various configurations of (n, d, B). The worst-case risks are obtained by taking a max of the worst-case risk estimate from DragonFly and the risks computed at randomly generated Σ's.

	Worst-case Risk							
	$\mathbf{n = 1.5 \times d, B = 1}$		$\mathbf{n = 1.5 \times d, B = 2}$		$\mathbf{n = 1.5 \times d, B = 4}$		$\mathbf{n = 1.5 \times d, B = 8}$	
Estimator	$\mathbf{d = 5}$	$\mathbf{d = 10}$	$\mathbf{d = 5}$	$\mathbf{d = 10}$	$\mathbf{d = 5}$	$\mathbf{d = 10}$	$\mathbf{d = 5}$	$\mathbf{d = 10}$
Empirical Covariance	2.5245	5.1095	2.5245	5.1095	2.5245	5.1095	2.5245	5.1095
James-Stein Estimator	2.1637	4.1704	2.1637	4.1704	2.1637	4.1704	2.1637	4.1704
Averaged Estimator	**1.8686**	**3.1910**	**1.9371**	**3.7019**	2.0827	4.2454	2.1416	**3.9864**

	Worst-case Risk							
	$\mathbf{n = 2 \times d, B = 1}$		$\mathbf{n = 2 \times d, B = 2}$		$\mathbf{n = 2 \times d, B = 4}$		$\mathbf{n = 2 \times d, B = 8}$	
Estimator	$\mathbf{d = 5}$	$\mathbf{d = 10}$	$\mathbf{d = 5}$	$\mathbf{d = 10}$	$\mathbf{d = 5}$	$\mathbf{d = 10}$	$\mathbf{d = 5}$	$\mathbf{d = 10}$
Empirical Covariance	1.8714	3.4550	1.8714	3.4550	1.8714	3.4550	1.8714	3.4550
James-Stein Estimator	1.6686	2.9433	1.6686	2.9433	1.6686	2.9433	1.6686	2.9433
Averaged Estimator	**1.2330**	**2.1944**	1.5237	**2.6471**	1.6050	3.0834	1.6500	2.9907

	Worst-case Risk							
	$\mathbf{n = 3 \times d, B = 1}$		$\mathbf{n = 3 \times d, B = 2}$		$\mathbf{n = 3 \times d, B = 4}$		$\mathbf{n = 3 \times d, B = 8}$	
Estimator	$\mathbf{d = 5}$	$\mathbf{d = 10}$	$\mathbf{d = 5}$	$\mathbf{d = 10}$	$\mathbf{d = 5}$	$\mathbf{d = 10}$	$\mathbf{d = 5}$	$\mathbf{d = 10}$
Empirical Covariance	1.1425	2.1224	1.1425	2.1224	1.1425	2.1224	1.1425	2.1224
James-Stein Estimator	1.0487	1.9068	1.0487	1.9068	1.0487	1.9068	**1.0487**	1.9068
Averaged Estimator	**0.8579**	**1.3731**	0.9557	**1.7151**	1.0879	1.9174	1.2266	2.0017

5.7.5 Entropy Estimation

In this section, we consider the problem of entropy estimation described in Section 5.2.4. Similar to covariance estimation, we use two layer neural networks to parameterize functions g, h in Equation (5.14). Implementing the minimization oracle then boils down to finding the parameters of these networks which minimize $\mathbb{E}_{P \sim P_t}\left[R(\hat{f}_{g,h}, P)\right]$. We use stochastic gradient descent to solve this optimization problem.

Baselines. We compare the performance of the estimators returned by Algorithm 5.1 for various values of (n, d), with the plugin MLE estimator $-\sum_{i=1}^{d} \hat{p}_i \log \hat{p}_i$, and the minimax rate optimal estimator of Jiao et al. [53] (JVHW). The plugin estimator is known to be sub-optimal in the high dimensional regime, where $n < d$ [53].

Results. We compare the performance of various estimators based on their worst-case risk computed using DragonFly. Since DragonFly is not guaranteed to compute the worst-case risk, we also compare the estimators based on their risk at randomly generated distributions (see Section 5.8.12). Table 5.6 presents the worst-case risk numbers. It can be seen that the plugin MLE estimator has a poor performance compared to JVHW and our estimator. Our estimator has similar performance as JVHW, which is the best known minimax estimator for entropy estimation. We believe the performance of our estimator can be improved with better hyper-parameters.

Table 5.6: Worst-case risk of various estimators for entropy estimation, for various values of (n, d). The worst-case risks are obtained by taking a max of the worst-case risk estimate from DragonFly and the risks computed at randomly generated distributions.

	Worst-case Risk									
	d = 10		**d = 20**		**d = 40**			**d = 80**		
Estimator	**n = 10**	**n = 20**	**n = 20**	**n = 40**	**n = 10**	**n = 20**	**n = 40**	**n = 20**	**n = 40**	**n = 80**
Plugin MLE	0.2895	0.1178	0.2512	0.0347	2.1613	0.8909	0.2710	2.2424	0.9142	0.2899
JVHW [53]	0.3222	0.0797	**0.1322**	0.0489	0.6788	0.2699	0.0648	**0.3751**	**0.1755**	0.0974
Averaged Estimator	**0.1382**	0.0723	0.1680	0.0439	**0.5392**	**0.2320**	0.0822	0.5084	0.2539	0.0672

5.8 Proofs

5.8.1 Measurability of Bayes Estimators

For any prior Π, define $p_\Pi(\mathbb{X}^n)$ as

$$\int_\theta \prod_{i=1}^n p(X_i;\theta) d\Pi(\theta).$$

For any prior Π, define estimator $\hat{\theta}_\Pi$ as follows

$$\hat{\theta}_\Pi(\mathbb{X}^n) \in \operatorname*{argmin}_{\tilde{\theta}\in\Theta} \mathbb{E}_{\theta\sim\Pi(\cdot|\mathbb{X}^n)} \left[M(\tilde{\theta},\theta) \right].$$

Certain regularity conditions need to hold for this to be a Bayes estimator of Π. $\hat{\theta}_\Pi$ defined this way need not be a measurable function of $\mathbb{X}^n$. We now provide sufficient conditions on the statistical problem which guarantee measurability of $\hat{\theta}_\Pi$. These conditions are from Brown and Purves [16].

Assumption 5.1. *The sample space $\mathcal{X}^n$ and the parameter set Θ are non-empty Borel sets.*

Assumption 5.2. *Let $\mathcal{B}(\mathcal{X}^n)$ be the Borel σ-algebra corresponding to the sample space $\mathcal{X}^n$ and $\mathcal{B}(\Theta)$ be the Borel σ-algebra corresponding to parameter space Θ. Let Π be a prior probability measure on Θ. Suppose, for each $\theta \in \Theta$, P_θ is such that, for each $B \in \mathcal{B}(\mathcal{X}^n)$, the function $\theta \to P_\theta(B)$ is measurable w.r.t $\mathcal{B}(\Theta)$.*

Assumption 5.3. *The loss function M defined on $\Theta \times \Theta$ and taking non-negative real values, is measurable w.r.t $\mathcal{B}(\Theta) \times \mathcal{B}(\Theta)$. Moreover, $M(\cdot,\theta)$ is lower semi-continuous on Θ, for each $\theta \in \Theta$.*

Under these assumptions, when Θ is compact, Brown and Purves [16] show that there exists a Borel measurable function $\hat{\theta}_\Pi$ such that

$$\hat{\theta}_\Pi(\mathbb{X}^n) \in \operatorname*{argmin}_{\tilde{\theta}\in\Theta} \mathbb{E}_{\theta\sim\Pi(\cdot|\mathbb{X}^n)} \left[M(\tilde{\theta},\theta) \right].$$

Moreover, $\hat{\theta}_\Pi$ is the Bayes estimator for Π.

5.8.2 Minimax Estimators, LFPs and Nash Equilibirium

Proposition 5.6. *Consider the statistical game in Equation (1.1). If $(\hat{\theta}^*, P^*)$ is a mixed strategy NE of (1.1), then the minmax and maxmin values of the linearized game are equal to each other. Moreover, $\hat{\theta}^*$ is a minimax estimator and P^* is an LFP. Conversely, if $\hat{\theta}^*$ is a minimax estimator, and P^* is an LFP, and the minmax and maxmin values of the linearized game (1.4) are equal to each other, then $(\hat{\theta}^*, P^*)$ is a mixed strategy NE of (1.1). Moreover, θ^* is a Bayes estimator for P^*.*

Proof. Suppose $(\hat{\theta}^*, P^*)$ is a mixed strategy NE. Then, from the definition of mixed strategy NE, we have

$$\sup_{P \in \mathcal{M}_\Theta} R(\hat{\theta}^*, P) \leq R(\hat{\theta}^*, P^*) \leq \inf_{\hat{\theta} \in \mathcal{M}_\mathcal{D}} R(\hat{\theta}, P^*).$$

This further implies

$$\inf_{\hat{\theta} \in \mathcal{M}_\mathcal{D}} \sup_{P \in \mathcal{M}_\Theta} R(\hat{\theta}, P) \overset{(a)}{\leq} \sup_{P \in \mathcal{M}_\Theta} R(\hat{\theta}^*, P) \leq R(\hat{\theta}^*, P^*) \overset{(b)}{\leq} \inf_{\hat{\theta} \in \mathcal{M}_\mathcal{D}} R(\hat{\theta}, P^*) \overset{(c)}{\leq} \sup_{P \in \mathcal{M}_\Theta} \inf_{\hat{\theta} \in \mathcal{M}_\mathcal{D}} R(\hat{\theta}, P).$$

Since $\inf_{\hat{\theta} \in \mathcal{M}_\mathcal{D}} \sup_{P \in \mathcal{M}_\Theta} R(\hat{\theta}, P) \geq \sup_{P \in \mathcal{M}_\Theta} \inf_{\hat{\theta} \in \mathcal{M}_\mathcal{D}} R(\hat{\theta}, P)$, the above set of inequalities all hold with an equality and imply that the minmax and maxmin values of the linearized game are equal to each other. Moreover, from (a), we have $\sup_{P \in \mathcal{M}_\Theta} R(\hat{\theta}^*, P) = \inf_{\hat{\theta} \in \mathcal{M}_\mathcal{D}} \sup_{P \in \mathcal{M}_\Theta} R(\hat{\theta}, P)$. This implies $\hat{\theta}^*$ is a minimax estimator. From (c), we have $\inf_{\hat{\theta} \in \mathcal{M}_\mathcal{D}} R(\hat{\theta}, P^*) = \sup_{P \in \mathcal{M}_\Theta} \inf_{\hat{\theta} \in \mathcal{M}_\mathcal{D}} R(\hat{\theta}, P)$. This implies P^* is an LFP. Finally, from (b), we have $R(\hat{\theta}^*, P^*) = \inf_{\hat{\theta} \in \mathcal{M}_\mathcal{D}} R(\hat{\theta}, P^*)$. This implies $\hat{\theta}^*$ is a Bayes estimator for P^*.

We now prove the converse. Since $\hat{\theta}^*$ is a minimax estimator and P^* is an LFP, we have

$$\sup_{P \in \mathcal{M}_\Theta} R(\hat{\theta}^*, P) = \inf_{\hat{\theta} \in \mathcal{M}_\mathcal{D}} \sup_{P \in \mathcal{M}_\Theta} R(\hat{\theta}, P), \quad \inf_{\hat{\theta} \in \mathcal{M}_\mathcal{D}} R(\hat{\theta}, P^*) = \sup_{P \in \mathcal{M}_\Theta} \inf_{\hat{\theta} \in \mathcal{M}_\mathcal{D}} R(\hat{\theta}, P).$$

Moreover, since minmax and maxmin values of the linearized game are equal to each other, all the above 4 quantities are equal to each other. Since $R(\hat{\theta}^*, P^*) \leq \sup_{P \in \mathcal{M}_\Theta} R(\hat{\theta}^*, P)$ and $R(\hat{\theta}^*, P^*) \geq \inf_{\hat{\theta} \in \mathcal{M}_\mathcal{D}} R(\hat{\theta}, P^*)$, we have

$$\sup_{P \in \mathcal{M}_\Theta} R(\hat{\theta}^*, P) = R(\hat{\theta}^*, P^*) = \inf_{\hat{\theta} \in \mathcal{M}_\mathcal{D}} R(\hat{\theta}, P^*).$$

This shows that $(\hat{\theta}^*, P^*)$ is a mixed strategy NE of the linear game in Equation (1.4). □

5.8.3 Follow the Perturbed Leader (FTPL)

We now describe the FTPL algorithm in more detail. We first introduce the notion of an offline optimization oracle, which takes as input a function $f : \mathcal{X} \to \mathbb{R}$ and a perturbation vector σ and returns an approximate minimizer of $f(\mathbf{x}) - \langle \sigma, \mathbf{x} \rangle$. An optimization oracle is called "(α, β)-approximate optimization oracle" if it returns $\mathbf{x}^* \in \mathcal{X}$ such that

$$f(\mathbf{x}^*) - \langle \sigma, \mathbf{x}^* \rangle \leq \inf_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}) - \langle \sigma, \mathbf{x} \rangle + \alpha + \beta \|\sigma\|_1.$$

Denote such an oracle with $\mathcal{O}_{\alpha,\beta}^{\text{FTPL}}(f,\sigma)$. Given access to such an oracle, the FTPL algorithm is given by the following prediction rule (see Algorithm 5.6)

$$\mathbf{x}_t = \mathcal{O}_{\alpha,\beta}^{\text{FTPL}}\left(\sum_{i=1}^{t-1} f_i, \sigma\right),$$

where $\sigma \in \mathbb{R}^d$ is a random perturbation such that $\{\sigma_j\}_{j=1}^d \overset{i.i.d}{\sim} \text{Exp}(\eta)$ and $\text{Exp}(\eta)$ is the exponential distribution with parameter η. We now state the following result from Suggala and Netrapalli [96] which provides an upper bound on the expected regret of Algorithm 5.6.

Theorem 5.11 (Regret Bound)**.** *Let D be the ℓ_∞ diameter of $\mathcal{X}$. Suppose the losses encountered by the learner are L-Lipschitz w.r.t ℓ_1 norm. For any fixed η, the predictions of Algorithm 5.6 satisfy the following regret bound*

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \frac{1}{T}\inf_{\mathbf{x}\in\mathcal{X}}\sum_{t=1}^{T} f_t(\mathbf{x})\right] \leq O\left(\eta d^2 D L^2 + \frac{d(\beta T + D)}{\eta T} + \alpha + \beta d L\right).$$

Algorithm 5.6 Follow the Perturbed Leader (FTPL)

1: **Input:** Parameter of exponential distribution η, approximate optimization subroutine $\mathcal{O}_{\alpha,\beta}$
2: **for** $t = 1 \dots T$ **do**
3: Generate random vector σ such that $\{\sigma_j\}_{j=1}^d \overset{i.i.d}{\sim} \text{Exp}(\eta)$
4: Predict $\mathbf{x}_t$ as

$$\mathbf{x}_t = \mathcal{O}_{\alpha,\beta}^{\text{FTPL}}\left(\sum_{i=1}^{t-1} f_i, \sigma\right).$$

5: Observe loss function f_t

5.8.4 Minimax Estimation via Online Learning

5.8.4.1 Proof of Proposition 5.1

We have the following bounds on the regret of the minimization and maximization players

$$\sum_{t=1}^{T} R(\hat{\theta}_t, P_t) - \inf_{\hat{\theta}\in\mathcal{D}} \sum_{t=1}^{T} R(\hat{\theta}, P_t) \leq \epsilon_1(T),$$
$$\sup_{\theta\in\Theta} \sum_{t=1}^{T} R(\hat{\theta}_t, \theta) - \sum_{t=1}^{T} R(\hat{\theta}_t, P_t) \leq \epsilon_2(T).$$

Now consider the following

$$\begin{aligned}
&\inf_{\hat{\theta}\in\mathcal{D}} \frac{1}{T}\sum_{t=1}^{T} R(\hat{\theta}, P_t) \\
&\geq \frac{1}{T}\sum_{t=1}^{T} R(\hat{\theta}_t, P_t) - \frac{\epsilon_1(T)}{T} \qquad (5.15)\\
&\geq \sup_{\theta\in\Theta} \frac{1}{T}\sum_{t=1}^{T} R(\hat{\theta}_t, \theta) - \frac{\epsilon_1(T) + \epsilon_2(T)}{T},
\end{aligned}$$

where the first and the second inequalities follow from the regret bounds of the minimization and maximization players. We further bound the LHS and RHS of the above inequality as follows

$$\inf_{\hat{\theta}\in\mathcal{D}} \frac{1}{T}\sum_{t=1}^{T} R(\hat{\theta}, P_t) \leq \frac{1}{T^2}\sum_{t=1}^{T}\sum_{t'=1}^{T} R(\hat{\theta}_{t'}, P_t) = R(\hat{\theta}_{RND}, P_{AVG}),$$

$$\sup_{\theta\in\Theta} \frac{1}{T}\sum_{t=1}^{T} R(\hat{\theta}_t, \theta) \geq \frac{1}{T^2}\sum_{t=1}^{T}\sum_{t'=1}^{T} R(\hat{\theta}_{t'}, P_t) = R(\hat{\theta}_{RND}, P_{AVG}).$$

Combining the previous two sets of inequalities gives us

$$R(\hat{\theta}_{RND}, P_{AVG}) \geq \sup_{\theta\in\Theta} R(\hat{\theta}_{RND}, \theta) - \frac{\epsilon_1(T)+\epsilon_2(T)}{T},$$

$$R(\hat{\theta}_{RND}, P_{AVG}) \leq \inf_{\hat{\theta}\in\mathcal{D}} R(\hat{\theta}, P_{AVG}) + \frac{\epsilon_1(T)+\epsilon_2(T)}{T}.$$

5.8.4.2 Proof of Theorem 5.1

To prove the Theorem we first bound the regret of each player and then rely on Proposition 5.1 to show that the iterates converge to a NE. Since the maximization player is responding using FTPL to the actions of minimization player, we rely on Theorem 5.11 to bound her regret. First note that the sequence of reward functions seen by the maximization player $R(\hat{\theta}_i, \cdot)$ are L-Lipschitz. Moreover, the domain Θ has ℓ_∞ diameter of D. So applying Theorem 5.11 gives us the following regret bound

$$\mathbb{E}_\sigma\left[\sup_{\theta\in\Theta}\frac{1}{T}\sum_{t=1}^{T} R(\hat{\theta}_t, \theta) - \frac{1}{T}\sum_{t=1}^{T} R(\hat{\theta}_t, \theta_t(\sigma))\right] \leq O\left(\eta d^2 D L^2 + \frac{d(\beta T + D)}{\eta T} + \alpha + \beta d L\right).$$

Taking the expectation inside, we get the following

$$\sup_{\theta\in\Theta}\frac{1}{T}\sum_{t=1}^{T} R(\hat{\theta}_t, \theta) - \frac{1}{T}\sum_{t=1}^{T} R(\hat{\theta}_t, P_t) \leq O\left(\eta d^2 D L^2 + \frac{d(\beta T + D)}{\eta T} + \alpha + \beta d L\right). \quad (5.16)$$

Since the minimization player is using BR, her regret is upper bounded by 0. Plugging in these two regret bounds in Proposition 5.1 gives us the required result.

5.8.4.3 Proof of Corollary 5.1

Note that this corollary is only concerned about existence of minimax estimators and LFPs, and showing that minmax and maxmin values of Equation (1.4) are equal to each other. So we can ignore the approximation errors introduced by the oracles and set $\alpha = \beta = \alpha' = 0$ in the results of Theorem 5.1 (that is, we assume access to exact optimization oracles, as we are only concerned with existence of NE and not about computational tractability of the algorithm).

Minimax Theorem. To prove the first part of the corollary, we set $\eta = \sqrt{\frac{1}{dL^2T}}$ in Theorem 5.1 and let $T \to \infty$. We get

$$\begin{aligned}
&\sup_{\theta \in \Theta} R(\hat{\theta}_{RND}, \theta) = \inf_{\hat{\theta} \in \mathcal{D}} R(\hat{\theta}, P_{AVG}) \\
&\implies \sup_{P \in \mathcal{M}_\Theta} R(\hat{\theta}_{RND}, P) = \inf_{\hat{\theta} \in \mathcal{M}_\mathcal{D}} R(\hat{\theta}, P_{AVG}) \\
&\implies \inf_{\hat{\theta} \in \mathcal{M}_\mathcal{D}} \sup_{P \in \mathcal{M}_\Theta} R(\hat{\theta}, P) \leq \sup_{P \in \mathcal{M}_\Theta} \inf_{\hat{\theta} \in \mathcal{M}_\mathcal{D}} R(\hat{\theta}, P).
\end{aligned}$$

Since minmax value of any game is always greater than or equal to maxmin value of the game, we get

$$\inf_{\hat{\theta} \in \mathcal{M}_\mathcal{D}} \sup_{P \in \mathcal{M}_\Theta} R(\hat{\theta}, P) = \sup_{P \in \mathcal{M}_\Theta} \inf_{\hat{\theta} \in \mathcal{M}_\mathcal{D}} R(\hat{\theta}, P) R^*.$$

Existence of LFP. We now show that the statistical game has an LFP. To prove this result, we make use of the following result on the compactness of probability spaces. If Θ is a compact space, then $\mathcal{M}_\Theta$ is sequentially compact; that is, any sequence $P_n \in \mathcal{M}_\Theta$ has a convergent subsequence converging to a point in $\mathcal{M}_\Theta$ (the notion of convergence here is weak convergence). Let $P_{AVG,t} = \frac{1}{t}\sum_{i=1}^{t} P_i$ be the mixture distribution obtained from the first t iterates of Algorithm 5.1 when run with $\eta = \sqrt{\frac{1}{dL^2T}}$ and exact optimization oracles. Consider the sequence of probability measures $\{P_{AVG,t}\}_{t=1}^{\infty}$. Since the parameter space Θ is compact, we know that there exists a converging subsequence $\{P_{AVG,t_i}\}_{i=1}^{\infty}$. Let $P^* \in \mathcal{M}_\Theta$ be the limit of this sequence. In the rest of the proof, we show that P^* is an LFP; that is, $\inf_{\hat{\theta} \in \mathcal{D}} R(\hat{\theta}, P^*) = R^*$. Since $R(\hat{\theta}, \theta)$ is bounded, and Lipschitz in its second argument, we have

$$\forall \hat{\theta} \in \mathcal{M}_\mathcal{D} \quad \lim_{i \to \infty} R(\hat{\theta}, P_{AVG,t_i}) = R(\hat{\theta}, P^*). \tag{5.17}$$

This follows from the equivalent formulations of weak convergence of measures. We now make use of the following result from Corollary 5.2 (which we prove later in Section 5.8.4.4)

$$\inf_{\hat{\theta} \in \mathcal{D}} R(\hat{\theta}, P_{AVG,t}) \geq R^* - O(t^{-\frac{1}{2}}).$$

Combining this with the fact that $\sup_{P \in \mathcal{M}_\Theta} \inf_{\hat{\theta} \in \mathcal{D}} R(\hat{\theta}, P) = R^*$, we get

$$\lim_{i \to \infty} \inf_{\hat{\theta} \in \mathcal{D}} R(\hat{\theta}, P_{AVG,t_i}) = R^*. \tag{5.18}$$

Equations (5.17), (5.18) show that $\inf_{\hat{\theta} \in \mathcal{D}} R(\hat{\theta}, P_{AVG,t_i})$, $R(\tilde{\theta}, P_{AVG,t_i})$ are converging sequences as $i \to \infty$. Since $\inf_{\hat{\theta} \in \mathcal{D}} R(\hat{\theta}, P_{AVG,t_i}) \leq R(\tilde{\theta}, P_{AVG,t_i})$ for all $i, \tilde{\theta} \in \mathcal{D}$, we have

$$\lim_{i \to \infty} \inf_{\hat{\theta} \in \mathcal{D}} R(\hat{\theta}, P_{AVG,t_i}) \leq \lim_{i \to \infty} R(\tilde{\theta}, P_{AVG,t_i}), \quad \forall \tilde{\theta} \in \mathcal{D}.$$

From Equations (5.17), (5.18), we then have

$$\begin{aligned}
&R^* \leq R(\tilde{\theta}, P^*), \quad \forall \tilde{\theta} \in \mathcal{D} \\
&\implies R^* \leq \inf_{\hat{\theta} \in \mathcal{D}} R(\hat{\theta}, P^*),
\end{aligned}$$

Combining this with the fact that $\sup_{P \in \mathcal{M}_\Theta} \inf_{\hat{\theta} \in \mathcal{D}} R(\hat{\theta}, P) = R^*$, we get

$$\inf_{\hat{\theta} \in \mathcal{D}} R(\hat{\theta}, P^*) = R^*.$$

This shows that P^* is an LFP.

Existence of Minimax Estimator. To show the existence of a minimax estimator, we make use of the following result from Wald [104], which is concerned about the "compactness" of the space of estimators $\mathcal{M}_\mathcal{D}$.

Proposition 5.7. *Suppose Θ is compact w.r.t $\Delta_M(\theta_1, \theta_2) = \sup_{\theta \in \Theta} |M(\theta_1, \theta) - M(\theta_2, \theta)|$. Moreover, suppose the risk R is bounded. Then for any sequence of $\{\hat{\theta}_i\}_{i=1}^\infty$ of estimators there exists a subsequence $\{\hat{\theta}_{i_j}\}_{j=1}^\infty$ such that $\lim_{j \to \infty} \hat{\theta}_{i_j} = \hat{\theta}_0$ and for any $\theta \in \Theta$*

$$\liminf_{i \to \infty} R(\hat{\theta}_{i_j}, \theta) \geq R(\hat{\theta}_0, \theta).$$

Let $\hat{\theta}_{RND,t}$ be the randomized estimator obtained by uniformly sampling an estimator from $\{\hat{\theta}_i\}_{i=1}^t$. Consider the sequence of estimators $\{\hat{\theta}_{RND,t}\}_{t=1}^\infty$. From the above proposition, we know that there exists a subsequence $\{\hat{\theta}_{RND,t_j}\}_{j=1}^\infty$ and an estimator $\hat{\theta}^*$ such that $\liminf_{j \to \infty} R(\hat{\theta}_{RND,t_j}, \theta) \geq R(\hat{\theta}^*, \theta)$. We now show that $\hat{\theta}^*$ is a minimax estimator; that is, we show that $\sup_{\theta \in \Theta} R(\hat{\theta}^*, \theta) = R^*$. We make use of the following result from Corollary 5.2

$$\sup_{\theta \in \Theta} R(\hat{\theta}_{RND,t}, \theta) \leq R^* + O(t^{-\frac{1}{2}}).$$

Combining this with the fact that $\inf_{\hat{\theta} \in \mathcal{D}} \sup_{P \in \mathcal{M}_\Theta} R(\hat{\theta}, P) = R^*$, we get

$$\lim_{j \to \infty} \sup_{\theta \in \Theta} R(\hat{\theta}_{RND,t_j}, \theta) = R^*. \tag{5.19}$$

Since $\sup_{\theta \in \Theta} R(\hat{\theta}_{RND,t_j}, \theta) \geq R(\hat{\theta}_{RND,t_j}, \tilde{\theta})$ for any $j, \tilde{\theta} \in \Theta$, we have

$$\liminf_{j \to \infty} \sup_{\theta \in \Theta} R(\hat{\theta}_{RND,t_j}, \theta) \geq \liminf_{j \to \infty} R(\hat{\theta}_{RND,t_j}, \tilde{\theta}) \geq R(\hat{\theta}^*, \theta), \quad \forall \tilde{\theta} \in \Theta.$$

Since $\{R(\hat{\theta}_{RND,t_j}, \theta)\}_{j=1}^\infty$ is a converging sequence, we have

$$\liminf_{j \to \infty} \sup_{\theta \in \Theta} R(\hat{\theta}_{RND,t_j}, \theta) = \lim_{j \to \infty} \sup_{\theta \in \Theta} R(\hat{\theta}_{RND,t_j}, \theta) = R^*.$$

This together with the previous inequality gives us $\sup_{\tilde{\theta} \in \Theta} R(\hat{\theta}_{RND,t_j}, \tilde{\theta}) \leq R^*$. This shows that θ^* is a minimax estimator.

5.8.4.4 Proof of Corollary 5.2

Minimax Estimator. From Theorem 5.1 we have

$$\begin{aligned}
\sup_{\theta\in\Theta} R(\hat{\theta}_{RND},\theta) &= \sup_{\theta\in\Theta}\frac{1}{T}\sum_{i=1}^{T} R(\hat{\theta}_i,\theta)\\
&\leq \inf_{\hat{\theta}\in\mathcal{D}}\frac{1}{T}\sum_{i=1}^{T} R(\hat{\theta},P_i) + O\left(\eta d^2DL^2 + \frac{d(\beta T+D)}{\eta T} + \alpha + \alpha' + \beta dL\right)\\
&= \inf_{\hat{\theta}\in\mathcal{M}_{\mathcal{D}}}\frac{1}{T}\sum_{i=1}^{T} R(\hat{\theta},P_i) + O\left(\eta d^2DL^2 + \frac{d(\beta T+D)}{\eta T} + \alpha + \alpha' + \beta dL\right)\\
&\overset{(a)}{\leq} \inf_{\hat{\theta}\in\mathcal{M}_{\mathcal{D}}}\sup_{P\in\mathcal{M}_\Theta} R(\hat{\theta},P) + O\left(\eta d^2DL^2 + \frac{d(\beta T+D)}{\eta T} + \alpha + \alpha' + \beta dL\right),
\end{aligned}$$

where (a) follows from the fact that $\sup_{\theta\in\Theta} R(\hat{\theta},\theta) \geq \frac{1}{T}\sum_{i=1}^{T} R(\hat{\theta},P_i)$. Substituting $\eta = \sqrt{\frac{1}{dL^2T}}$ in the above equation shows that the randomized estimator is approximately minimax. This completes the first part of the proof. If the metric M is convex in its first argument, then from Jensen's inequality we have

$$\forall\theta, \quad R(\hat{\theta}_{AVG},\theta) \leq R(\hat{\theta}_{RND},\theta).$$

This shows that the worst-case risk of $\hat{\theta}_{AVG}$ is upper bounded as

$$\sup_{\theta\in\Theta} R(\hat{\theta}_{AVG},\theta) \leq \inf_{\hat{\theta}\in\mathcal{M}_{\mathcal{D}}}\sup_{P\in\mathcal{M}_\Theta} R(\hat{\theta},\theta) + O\left(\eta d^2DL^2 + \frac{d(\beta T+D)}{\eta T} + \alpha + \alpha' + \beta dL\right). \tag{5.20}$$

Substituting $\eta = \sqrt{\frac{1}{dL^2T}}$ in Equation (5.20) gives us the required bound on the worst-case risk of $\hat{\theta}_{AVG}$.

LFP. We now prove the results pertaining to LFP. From Theorem 5.1, we have

$$\begin{aligned}
\inf_{\hat{\theta}\in\mathcal{M}_{\mathcal{D}}} R(\hat{\theta},P_{AVG}) &= \inf_{\hat{\theta}\in\mathcal{M}_{\mathcal{D}}}\frac{1}{T}\sum_{i=1}^{T} R(\hat{\theta},P_i)\\
&\geq \sup_{P\in\mathcal{M}_\Theta}\frac{1}{T}\sum_{i=1}^{T} R(\hat{\theta}_i,P) - O\left(\eta d^2DL^2 + \frac{d(\beta T+D)}{\eta T} + \alpha + \alpha' + \beta dL\right)\\
&\geq \inf_{\hat{\theta}\in\mathcal{M}_{\mathcal{D}}}\sup_{P\in\mathcal{M}_\Theta} R(\hat{\theta},P) - O\left(\eta d^2DL^2 + \frac{d(\beta T+D)}{\eta T} + \alpha + \alpha' + \beta dL\right).
\end{aligned}$$

Substituting $\eta = \sqrt{\frac{1}{dL^2T}}$ in the above equation shows that P_{AVG} is approximately least favourable. Now consider the case where M is convex in its first argument. To show that $\hat{\theta}_{AVG}$ is an approximate Bayes estimator for P_{AVG}, we again rely on Theorem 5.1 where we showed that

$$\sup_{P\in\mathcal{M}_\Theta}\frac{1}{T}\sum_{i=1}^{T} R(\hat{\theta}_i,P) \leq \inf_{\hat{\theta}\in\mathcal{M}_{\mathcal{D}}}\frac{1}{T}\sum_{t=1}^{T} R(\hat{\theta},P_t) + O\left(\eta d^2DL^2 + \frac{d(\beta T+D)}{\eta T} + \alpha + \alpha' + \beta dL\right).$$

Since $\frac{1}{T^2}\sum_{t=1}^{T}\sum_{t'=1}^{T} R(\hat{\theta}_{t'}, P_t) \leq \sup_{P\in\mathcal{M}_\Theta} \frac{1}{T}\sum_{i=1}^{T} R(\hat{\theta}_i, P)$, we have

$$\frac{1}{T^2}\sum_{t=1}^{T}\sum_{t'=1}^{T} R(\hat{\theta}_{t'}, P_t) \leq \inf_{\hat{\theta}\in\mathcal{M}_\mathcal{D}} \frac{1}{T}\sum_{t=1}^{T} R(\hat{\theta}, P_t) + O\left(\eta d^2 D L^2 + \frac{d(\beta T + D)}{\eta T} + \alpha + \alpha' + \beta d L\right).$$

Since M is convex in its first argument, we have

$$\frac{1}{T^2}\sum_{t=1}^{T}\sum_{t'=1}^{T} R(\hat{\theta}_{t'}, P_t) \geq \frac{1}{T}\sum_{i=1}^{T} R(\hat{\theta}_{AVG}, P_i).$$

Combining the above two equations shows that $\hat{\theta}_{AVG}$ is an approximate Bayes estimator for P_{AVG}.

5.8.5 Invariance of Minimax Estimators

5.8.5.1 Proof of Theorem 5.2

In our proof, we rely on the following property of left Haar measure μ of a compact group G. For any real valued integrable function f on G and any $g \in G$ [see Chapter 7 of 106]

$$\int_G f(g^{-1}h)d\mu(h) = \int_G f(h)d\mu(h). \tag{5.21}$$

We now proceed to the proof of the Theorem. For any estimator $\hat{\theta} : \mathcal{X}^n \to \Theta$, define the following estimator $\hat{\theta}_G$

$$\hat{\theta}_G(\mathbb{X}^n) = \int_G g\hat{\theta}(g^{-1}\mathbb{X}^n)d\mu(g),$$

where μ is the left Haar measure on G and $g\mathbb{X}^n = \{gX_1, \ldots gX_n\}$. The above integral is well defined because $\hat{\theta}$ is measurable, G is compact and the action of the group G is continuous. We first show that $\hat{\theta}_G$ is invariant under group transformations G. For any $h \in G$, consider the following

$$\begin{aligned}
\hat{\theta}_G(h\mathbb{X}^n) &= \int_G g\hat{\theta}((g^{-1}h)\mathbb{X}^n)d\mu(g) \\
&= \int_G h(h^{-1}g)\hat{\theta}((h^{-1}g)^{-1}\mathbb{X}^n)d\mu(g) \\
&= h\left[\int_G (h^{-1}g)\hat{\theta}((h^{-1}g)^{-1}\mathbb{X}^n)d\mu(g)\right] \\
&\overset{(a)}{=} h\left[\int_G g\hat{\theta}(g^{-1}\mathbb{X}^n)d\mu(g)\right] \\
&= h\hat{\theta}_G(\mathbb{X}^n),
\end{aligned}$$

where (a) follows from Equation (5.21). This shows that $\hat{\theta}_G$ is an invariant estimator. We now show that the worst case risk of $\hat{\theta}_G$ is less than or equal to the worst case risk of $\hat{\theta}$.

Consider the following upper bound on the risk of $\hat{\theta}_G$ at any $\theta \in \Theta$

$$\begin{aligned}
R(\hat{\theta}_G, \theta) &= \mathbb{E}_{\mathbb{X}^n \sim P_\theta^n}\left[M(\hat{\theta}_G(\mathbb{X}^n), \theta)\right] \\
&\leq \mathbb{E}_{\mathbb{X}^n \sim P_\theta^n}\left[\int_G M(g\hat{\theta}(g^{-1}\mathbb{X}^n), \theta) d\mu(g)\right] \quad \text{(convexity of } M\text{)} \\
&= \mathbb{E}_{\mathbb{X}^n \sim P_\theta^n}\left[\mathbb{E}_{g\sim\mu}\left[M(g\hat{\theta}(g^{-1}\mathbb{X}^n), \theta)\right]\right] \\
&\overset{(a)}{=} \mathbb{E}_{g\sim\mu}\left[\mathbb{E}_{\mathbb{X}^n \sim P_{g^{-1}\theta}^n}\left[M(g\hat{\theta}(\mathbb{X}^n), \theta)\right]\right] \quad \text{(change of variables)} \\
&\overset{(b)}{=} \mathbb{E}_{g\sim\mu}\left[\mathbb{E}_{\mathbb{X}^n \sim P_{g^{-1}\theta}^n}\left[M(\hat{\theta}(\mathbb{X}^n), g^{-1}\theta)\right]\right] \quad \text{(invariance of } M\text{)} \\
&= \mathbb{E}_{g\sim\mu}\left[R(\hat{\theta}, g^{-1}\theta)\right] \\
&\leq \sup_{\theta' \in \Theta} R(\hat{\theta}, \theta'),
\end{aligned}$$

where (a) follows from Fubini's theorem and change of variables $X' = g^{-1}X$ and the fact that if $X \sim P_\theta$, then $g^{-1}X \sim P_{g^{-1}\theta}$. (b) follows from the invariance property of the metric M. This shows that $\sup_{\theta\in\Theta} R(\hat{\theta}_G, \theta) \leq \sup_{\theta\in\Theta} R(\hat{\theta}, \theta)$. This shows that we can always improve a given estimator by averaging over the group G and hence there should be a minimax estimator which is invariant under the action of G.

5.8.5.2 Proof of Theorem 5.3

We first prove some intermediate results which we require in the proof of Theorem 5.3.

Lemma 5.1. *Suppose $\hat{\theta}$ is a deterministic estimator that is invariant to group transformations G. Then $R(\hat{\theta}, \theta_1) = R(\hat{\theta}, \theta_2)$, whenever $\theta_1 \sim \theta_2$.*

Proof. Suppose $\theta_2 = g\theta_1$ for some $g \in G$. From the definition of $R(\hat{\theta}, g\theta_1)$ we have

$$\begin{aligned}
R(\hat{\theta}, \theta_2) = R(\hat{\theta}, g\theta_1) &= \mathbb{E}_{\mathbb{X}^n \sim P_{g\theta_1}^n}\left[M(\hat{\theta}(\mathbb{X}^n), g\theta_1)\right] \\
&= \mathbb{E}_{\mathbb{X}^n \sim P_{g\theta_1}^n}\left[M(g^{-1}\hat{\theta}(\mathbb{X}^n), \theta_1)\right] \quad \text{(invariance of loss metric)} \\
&= \mathbb{E}_{\mathbb{X}^n \sim P_{g\theta_1}^n}\left[M(\hat{\theta}(g^{-1}\mathbb{X}^n), \theta_1)\right] \quad \text{(invariance of estimator)} \\
&\overset{(a)}{=} \mathbb{E}_{\mathbb{X}^n \sim P_{\theta_1}^n}\left[M(\hat{\theta}(\mathbb{X}^n), \theta_1)\right] \\
&= R(\hat{\theta}, \theta_1),
\end{aligned}$$

where (a) follows from the fact that $gX \sim P_{g\theta}$ whenever $X \sim P_\theta$. This shows that $R(\hat{\theta}, \theta_1) = R(\hat{\theta}, \theta_2)$. □

Lemma 5.2. *Suppose Π is a probability distribution which is invariant to group transformations G. For any deterministic estimator $\hat{\theta}$, there exists an invariant estimator $\hat{\theta}_G$ such that the Bayes risk of $\hat{\theta}_G$ is no larger than the Bayes risk of $\hat{\theta}$*

$$R(\hat{\theta}, \Pi) \geq R(\hat{\theta}_G, \Pi).$$

Proof. Define estimator $\hat{\theta}_G$ as follows

$$\hat{\theta}_G(\mathbb{X}^n) = \int_G g\hat{\theta}(g^{-1}\mathbb{X}^n)d\mu(g),$$

where μ is the left Haar measure on G. Note that, in the proof of Theorem 5.2 we showed that this estimator is invariance to the action of group G. We now show that the Bayes risk of $\hat{\theta}_G$ is less than equal to the Bayes risk of $\hat{\theta}$. Consider the following

$$\begin{aligned}
R(\hat{\theta}_G, \Pi) &= \mathbb{E}_{\theta\sim\Pi}[R(\hat{\theta}_G, \theta)] \\
&= \mathbb{E}_{\theta\sim\Pi}\left[\mathbb{E}_{\mathbb{X}^n\sim P_\theta^n}\left[M\left(\int_G g\hat{\theta}(g^{-1}\mathbb{X}^n)d\mu(g), \theta\right)\right]\right] \\
&\overset{(a)}{\leq} \mathbb{E}_{\theta\sim\Pi}\left[\mathbb{E}_{\mathbb{X}^n\sim P_\theta^n}\left[\mathbb{E}_{g\sim\mu}\left[M\left(g\hat{\theta}(g^{-1}\mathbb{X}^n), \theta\right)\right]\right]\right] \\
&= \mathbb{E}_{g\sim\mu}\left[\mathbb{E}_{\theta\sim\Pi}\left[\mathbb{E}_{\mathbb{X}^n\sim P_\theta^n}\left[M\left(g\hat{\theta}(g^{-1}\mathbb{X}^n), \theta\right)\right]\right]\right] \\
&\overset{(b)}{=} \mathbb{E}_{g\sim\mu}\left[\mathbb{E}_{\theta\sim\Pi}\left[\mathbb{E}_{\mathbb{X}^n\sim P_\theta^n}\left[M\left(\hat{\theta}(g^{-1}\mathbb{X}^n), g^{-1}\theta\right)\right]\right]\right] \\
&= \mathbb{E}_{g\sim\mu}\left[\mathbb{E}_{\theta\sim\Pi}\left[R(\hat{\theta}, g^{-1}\theta)\right]\right] \\
&\overset{(c)}{=} \mathbb{E}_{\theta\sim\Pi}\left[R(\hat{\theta}, \theta)\right],
\end{aligned}$$

where (a) uses convexity of M and follows from Jensen's inequality, (b) follows from the invariance of M and (c) follows from the invariance of distribution Π to actions of group G. □

We now proceed to the proof of Theorem 5.3. We first prove the second part of the Theorem. The first part immediately follows from the proof of second part. Suppose $(\hat{\theta}_G^*, P_G^*)$ is an ϵ-approximate mixed strategy Nash equilibirium of the reduced statistical game in Equation (5.3). Our goal is to construct an approximate Nash equilibrium of the original statistical game in Equation (1.1), using $(\hat{\theta}_G^*, P_G^*)$.

Note that $\hat{\theta}_G^*$ is a randomized estimator over the set of deterministic invariant estimators $\mathcal{D}_G$ and P_G^* is a distribution on the quotient space Θ/G. To construct an approximate Nash equilibrium of the original statistical game (1.1), we extend P_G^* to the entire parameter space Θ. We rely on Bourbaki's approach to measure theory, which is equivalent to classical measure theory in the setting of locally compact spaces we consider in this work [106]. In Bourbaki's approach, any measure ν on a set Θ is defined as a linear functional on the set of integrable functions (that is, a measure is defined by its action on integrable functions)

$$\nu[f] = \int_\Theta f(\theta)d\nu(\theta).$$

We define P^*, the extension of P_G^* to the entire parameter space Θ, as follows

$$P^*[f] = \int_{\Theta/G} f'(\Theta_\beta)dP_G^*(\Theta_\beta),$$

where $f' : \Theta/G \to \mathbb{R}$ is a function that depends on f, and is defined as follows. First define $f_I : \Theta \to \mathbb{R}$, an invariant function constructed using f, as $f_I(\theta) = \int_\Theta f(g\theta)d\mu(g)$, where

μ is the left invariant Haar measure of G. From Equation (5.21), it is easy to see that $f_I(h\theta) = f_I(\theta)$, for all $h \in G$. So f_I is constant on the equivalence classes of Θ. So f_I can be written in terms of a function $f' : \Theta/G \to \mathbb{R}$, as follows

$$f_I = f' \circ \gamma,$$

where $\gamma : \Theta \to \Theta/G$ is the orbit projection function which projects $\theta \in \Theta$ onto the quotient space. We first show that P^* defined this way is an invariant measure. To this end, we use the following equivalent definition of an invariant measure.

Proposition 5.8. *A probability measure ν on Θ is invariant to transformations of group G iff for any ν-integrable function f and for any $h \in G$, $\int f(\theta)d\nu(\theta) = \int f(h\theta)d\nu(\theta)$.*

Since f_I is an invariant function, relying on the above proposition, it is easy to see that P^* is an invariant measure. We now show that $(\hat{\theta}^*_G, P^*)$ is an ϵ-approximate mixed strategy Nash equilibrium of Equation (1.1). Since $(\hat{\theta}^*_G, P^*_G)$ is an ϵ-approximate Nash equilibrium of Equation (5.3), we have

$$\sup_{\Theta_\beta \in \Theta/G} R_G(\hat{\theta}^*_G, \Theta_\beta) - \epsilon \leq \mathbb{E}_{\Theta_\beta \sim P^*_G}[R_G(\hat{\theta}^*_G, \Theta_\beta)] \leq \inf_{\hat{\theta} \in \mathcal{D}_G} \mathbb{E}_{\Theta_\beta \sim P^*_G}[R_G(\hat{\theta}, \Theta_\beta)] + \epsilon, \quad (5.22)$$

where $\mathcal{D}_G$ is the set of deterministic invariant estimators. Now consider the following

$$\begin{aligned}
\mathbb{E}_{\theta \sim P^*}[R(\hat{\theta}^*_G, \theta)] &\overset{(a)}{=} \mathbb{E}_{\Theta_\beta \sim P^*_G}[R_G(\hat{\theta}^*_G, \Theta_\beta)] \quad \text{(Lemma 5.1)} \\
&\leq \inf_{\hat{\theta} \in \mathcal{D}_G} \mathbb{E}_{\Theta_\beta \sim P^*_G}[R_G(\hat{\theta}, \Theta_\beta)] + \epsilon \quad \text{(Equation (5.22))} \\
&= \inf_{\hat{\theta} \in \mathcal{D}_G} \mathbb{E}_{\theta \sim P^*}[R(\hat{\theta}, \theta)] + \epsilon \quad \text{(definition of } P^*) \\
&\overset{(b)}{=} \inf_{\hat{\theta} \in \mathcal{D}} \mathbb{E}_{\theta \sim P^*}[R(\hat{\theta}, \theta)] + \epsilon \quad \text{(Lemma 5.2)},
\end{aligned}$$

where (a) follows from the definition of P^* and Lemma 5.1. (b) follows from the fact that for any invariant prior, there exists a Bayes estimator which is invariant to group transformations (Lemma 5.2). Next, we provide a lower bound for $\mathbb{E}_{\theta \sim P^*}[R(\hat{\theta}^*_G, \theta)]$

$$\begin{aligned}
\mathbb{E}_{\theta \sim P^*}[R(\hat{\theta}^*_G, \theta)] &= \mathbb{E}_{\Theta_\beta \sim P^*_G}[R_G(\hat{\theta}^*_G, \Theta_\beta)] \\
&\geq \sup_{\Theta_\beta \in \Theta/G} R_G(\hat{\theta}^*_G, \Theta_\beta) - \epsilon \\
&= \sup_{\theta \in \Theta} R(\hat{\theta}^*_G, \theta) - \epsilon \quad \text{(Lemma 5.1)}
\end{aligned}$$

The upper and lower bounds for $\mathbb{E}_{\theta \sim P^*}[R(\hat{\theta}^*_G, \theta)]$ derived in the previous two equations shows that $(\hat{\theta}^*_G, P^*)$ is an ϵ-approximate mixed strategy Nash equilibrium of the original statistical game in Equation 1.1. The above inequalites also show that

$$\sup_{\theta \in \Theta} R(\hat{\theta}^*_G, \theta) - \epsilon \leq \mathbb{E}_{\Theta_\beta \sim P^*_G}[R_G(\hat{\theta}^*_G, \Theta_\beta)] \leq \inf_{\hat{\theta} \in \mathcal{D}} \mathbb{E}_{\theta \sim P^*}[R(\hat{\theta}, \theta)] + \epsilon.$$

This, together with Equation (5.22), shows that

$$\inf_{\hat{\theta} \in \mathcal{M}_\mathcal{D}} \sup_{\theta \in \Theta} R(\hat{\theta}, \theta) = \inf_{\hat{\theta} \in \mathcal{M}_{\mathcal{D},G}} \sup_{\Theta_\beta \in \Theta/G} R_G(\hat{\theta}, \Theta_\beta).$$

5.8.6 Applications of Invariance Theorem

In our proofs, we establish homeomorphisms between the quotient spaces and another natural space over which we run our algorithm. Note that establishing a homeomorphism is sufficient since we are only dealing with Borel σ-algebras on our spaces and homeomorphism would imply that there is an isomorphism between the Borel σ-algebras of the two spaces. Hence, measures learnt on one space can be transferred to another.

5.8.6.1 Proof of Theorem 5.4

First note that for any $g \in \mathbb{O}(d)$ and $\theta \in \Theta$, we have $g\theta \in \Theta$ and the distribution of gX is $P_{g\theta}$. Moreover, for any orthogonal matrix $g \in \mathbb{O}(d)$ we have $\|g\theta - gX\|^2 = \|\theta - X\|^2$, which implies the statistical game is invariant to group transformations G.

For the second part, note that for any $\theta_1, \theta_2 \in \Theta$ such that $\|\theta_1\|_2 = \|\theta_2\|_2$, $\exists g \in \mathbb{O}(d)$ s.t. $g\theta_1 = \theta_2$. Mapping all elements to their norm gives us a bijection between the quotient space and the interval $[0, B]$. The continuity of this bijection and it's inverse can easily be checked using the standard basis for both the topologies.

5.8.6.2 Proof of Theorem 5.5

Note that for any $\theta \in \Theta$, $g\theta = [g_1\theta^{1:k},\ g_2\theta^{k+1:d}] \in \Theta$. Since g_1 is orthogonal, for any $\theta_1, \theta_2 \in \Theta$ we have $\|g_1\theta_1^{1:k} - g_1\theta_2^{1:k}\| = \|\theta_1^{1:k} - \theta_2^{1:k}\|$. Hence the invariance of the statistical game follows.

Now, for any $\theta_1, \theta_2 \in \Theta$ such that $\|\theta_1^{1:k}\| = \|\theta_2^{1:k}\|$ and $\|\theta_1^{k+1:d}\| = \|\theta_2^{k+1:d}\|$, $\exists g_1 \in \mathbb{O}(k)$ and $g_2 \in \mathbb{O}(d-k)$ such that $g_1\theta_1^{1:k} = \theta_2^{1:k}$ and $g_2\theta_1^{k+1:d} = \theta_2^{k+1:d}$. Hence $\exists g \in \mathbb{O}(k) \times \mathbb{O}(d-k)$ such that $g\theta_1 = \theta_2$. This means that in each equivalence class the parameters $B_1 = \|\theta_1^{1:k}\|^2$ and $B_2 = \|\theta_1^{k+1:d}\|^2$ are constant. Since $\|\theta\|^2 \leq B$ we have $B_1 + B_2 \leq B$, this gives us a bijection. The continuity of this bijection and it's inverse can easily be checked using the standard basis for both the topologies.

5.8.6.3 Proof of Theorem 5.6

We define the action of any $g \in \mathbb{O}(d)$ on the samples $\{(X_i, Y_i)\}_{i=1}^n$ as transforming them to $\{(gX_i, Y_i)\}_{i=1}^n$. Since $Y_i = X_i^T\theta + \epsilon_i = X_i^T g^T g\theta + \epsilon_i = (gX_i)^T g\theta + \epsilon_i$ and $\|g\theta_1 - g\theta_2\| = \|\theta_1 - \theta_2\|$ for any $\theta_1, \theta_2 \in \Theta$ we have the invariance of the statistical game. The rest of the proof uses similar arguments as in Theorem 5.4.

5.8.6.4 Proof of Theorem 5.7

First note that for any Σ such that $\|\Sigma\|_2 \leq B$, and any $g \in \mathbb{O}(d)$, we have $\|g\Sigma g^T\| \leq B$. If $X \sim N(0, \Sigma)$ then for any $g \in \mathbb{O}(d)$

$$\mathbb{E}[gXX^Tg^T] = g\mathbb{E}[XX^T]g^T = g\Sigma g^T.$$

Hence $gX \sim N(0, g\Sigma g^T)$. Moreover, we have

$$\begin{aligned}
&M(g\Sigma_1 g^T, g\Sigma_2 g^T) \\
&= \operatorname{tr}\left((g\Sigma_1 g^T)^{-1} g\Sigma_2 g^T\right) - \log|(g\Sigma_1 g^T)^{-1} g\Sigma_2 g^T| - d \\
&= \operatorname{tr}\left(g\Sigma_1^{-1} g^T g\Sigma_2^{-1} g^T\right) - \log|g\Sigma_1^{-1} g^T g\Sigma_2^{-1} g^T| - d \\
&= \operatorname{tr}(g\Sigma_1^{-1}\Sigma_2 g^T) - \log|g\Sigma_1^{-1}\Sigma_2 g^T| - d \\
&= M(\Sigma_1, \Sigma_2),
\end{aligned}$$

where the last equality follows from the invariance of trace to multiplication with orthogonal matrices and the property of the determinant to split over the multiplication of matrices. This shows the desired invariance of the statistical game.

Now, consider two covariance matrices Σ_1, Σ_2 with singular value decompositions (SVD) $\Sigma_1 = U_1\Delta_1 U_1^T$ and $\Sigma_2 = U_2\Delta_2 U_2^T$ respectively. Here all matrices are square and of full rank. In particular, Δ_1 and Δ_2 are diagonal matrices with decreasing entries from left to right and, U_1 and U_2 are orthogonal matrices. Since the orthogonal group is transitive $\exists g \in \mathbb{O}(d)$ such that $gU_1 = U_2$. If $\Delta_1 = \Delta_2$ we have $g\Sigma_1 g^T = \Sigma_2$. Hence under the action of $\mathbb{O}(d)$, all covariance matrices with the same singular values fall in the same equivalence class. It is easy to see that this is also a necessary condition. These equivalence classes naturally form a bijection with a sequence of d decreasing positive real numbers bounded above by B. The continuity of this bijection and it's inverse can easily be checked using the standard basis for both the topologies.

5.8.6.5 Proof of Theorem 5.8

Let P, Q be any two distributions on d elements $\{1, \dots d\}$ such that $\exists g \in S_d$ s.t. $gP = Q$. They are indistinguishable from the samples they generate. Since the entropy is defined as

$$f(P) = -\sum_{i=1}^{d} p_i \log(p_i)$$

it doesn't depend upon the ordering of the individual probabilites. Hence the statistical game is invariant under the action of S_d.

Since using a permutation we can always order a given set of probabilities in decreasing order, there is a natural bijection between the quotient space and the given space. The continuity of this map and it's inverse can easily be checked using the standard basis for both the topologies.

5.8.6.6 Mixture of Gaussians

In the problem of mixture of Gaussians we are given n samples $X_1, \dots, X_n \in \mathbb{R}^d$ which come from a mixture distribution of k Gaussians with different means

$$P_\theta = \sum_{i=1}^{k} p_i \mathcal{N}(\theta_i, \Sigma_i).$$

We assume that all k Gaussians have the same covariance, let's say identity, and we also assume that we know the mixture probabilities. Finally, we assume that the mean vectors

θ_i are such that $\|\theta_i\| \leq B$. Under this setting we want to estimate the k different means while minimizing the sum of the L_2^2 losses of all the estimates of the mean parameters.

We will show the invariance of this statistical game under the action of the group $G = \mathbb{O}(d) \times \mathbb{O}(d-1) \times \ldots \times \mathbb{O}(d-k+1)$. But first we describe an element in the group and it's operation on the parameter and sample space.

An element of $g \in G$ is made up of a sequence of k orthonormal matrices $(g_1, \ldots, g_k)$ such that for a given set of parameters $\theta = (\theta_1, \ldots, \theta_k) \in \mathbb{R}^{d \times k}$ (where each $\theta_i \in \mathbb{R}^d$) the matrix g_i leaves the first $(i-1)$ parameters unchanged, i.e. for $j = 1, \ldots, i-1$ $g_i \theta_j = \theta_j$. Hence the ith orthonormal matrix has $(d-i+1)$ degrees of freedom and can be viewed as an element in $\mathbb{O}(d-i+1)$.

The action of g on θ is defined as

$$\begin{aligned} g\theta &= g(\theta_1, \ldots, \theta_k) \\ &= (g\theta_1, \ldots, g\theta_k) \\ &= (g_k \ldots g_1 \theta_1, \ldots, g_k \ldots g_1 \theta_1) \\ &= (g_1 \theta_1, \ldots, g_i \ldots g_1 \theta_i, \ldots, g_k \ldots g_1 \theta_k) \end{aligned}$$

where the last equality follows from the definition of our group. The group acts in a similar manner on the sample space, i.e., for an $X \in \mathcal{X}$ $gX = g_k \ldots g_1 X$.

Theorem 5.12. *The statistical game defined by mixture of k-Gaussians with identity covariance and known mixture probabilities under L_2^2 loss is invariant under the action of the group $\mathbb{O}(d) \times \mathbb{O}(d-1) \times \ldots \times \mathbb{O}(d-k+1)$. Moreover, the quotient space is homeomorphic to $(0, B]^k \times [0, \pi]^{\binom{k}{2}}$.*

Proof. First we show the invariance of the mixture distribution $P_\theta = \sum_i p_i \mathcal{N}(\theta_i, I)$, i.e., if $X \sim P_\theta$ then $gX \sim P_{g\theta}$. Note that from the proof of Theorem 5.4 it follows that for a given normal distribution $N(\tilde{\theta}, I)$ and an orthonormal matrix $h \in \mathbb{O}(d)$ s.t. $h\tilde{\theta} = \tilde{\theta}$ if $X \sim N(\tilde{\theta}, I)$ then $hX \sim N(h\tilde{\theta}, I) = N(\tilde{\theta}, I)$. The invariance of P follows directly from this by substituting each $\|X - \theta_i\|^2$ in the pdf with $\|g_k \ldots g_1 X - g_k \ldots g_1 \theta_i\|^2$ and the definition of the group. The L_2^2 loss is trivially invariant and hence we establish the invariance of the statistical game.

Now, notice that for any two given parameters $\theta = (\theta_1, \ldots, \theta_k), \phi = (\phi_1, \ldots, \phi_k) \in \mathbb{R}^{dk}$ if we have the property that $\forall i \ \|\theta_i\| = \|\phi_i\|$ and $\forall i, j \ \theta_i^T \theta_j = \phi_i^T \phi_j$ then we can find orthonormal matrices $g_1, \ldots, g_k$ s.t. $\forall i \ g_i \ldots g_1 \theta_i = \phi_i$. This follows from the following inductive argument: Assume we have $g_1, \ldots, g_{i-1}$ which satisfy the given constraints. Consider $\theta' = g_{i-1} \ldots g_1 \theta_i$. We have $\forall j = 1, \ldots, i-1 \ \theta'^T \phi_j = \theta_i^T \theta_j = \phi_i^T \phi_j$ because $g^T = g^{-1}$. Now if ϕ_i lies in the span of $\phi_1, \ldots, \phi_{i-1}$ then $\theta' = \phi_i$ and we can pick g_i to be any orthonormal matrix which doesn't transform this spanned space. Otherwise, we can pick an orthonormal matrix which rotates the axis orthogonal to the spanned subspace and in the direction of the high component of θ' to the corresponding axis for ϕ_i. This completes the desired construction.

It is easy to see that given θ, ϕ, g which satisfy $g\theta = \phi$, we have $\forall i \ \|\theta_i\| = \|\phi_i\|$ and $\forall i, j \ \theta_i^T \theta_j = \phi_i^T \phi_j$. Hence the equivalence classes are defined uniquely by the norms of the

individual gaussians and the angles between them, since there are k different norms and $\binom{k}{2}$ many angles we can establish a bijection between the quotient space and $(0, B]^k \times [0, \pi]^{\binom{k}{2}}$. The continuity of this map and it's inverse can easily be checked using the standard basis for both the topologies. □

5.8.7 Finite Gaussian Sequence Model

5.8.7.1 Proof of Proposition 5.2

In this section we derive a closed-form expression for the minimizer $\hat{\theta}_t$ of the following objective

$$\operatorname*{argmin}_{\hat{\theta} \in \mathcal{D}_G} \mathbb{E}_{b \sim P_t} \left[R(\hat{\theta}, b\mathbf{e}_1) \right],$$

where $\mathcal{D}_G$ is the set of deterministic estimators which are invariant to transformations of orthogonal group $\mathbb{O}(d)$. From Lemma 5.1, we know that for any invariant estimator $\hat{\theta} \in \mathcal{D}_G$ and any $g \in \mathbb{O}(d)$, $R(\hat{\theta}, b\mathbf{e}_1) = R(\hat{\theta}, bg\mathbf{e}_1)$. So the above problem can be rewritten as follows

$$\operatorname*{argmin}_{\hat{\theta} \in \mathcal{D}_G} \mathbb{E}_{b \sim P_t} \left[\mathbb{E}_{\theta \sim U_b} \left[R(\hat{\theta}, \theta) \right] \right],$$

where U_b is the uniform distribution over spherical shell of radius b, centered at origin; that is, its density $u_b(\theta)$ is defined as

$$u_b(\theta) \propto \begin{cases} 0, & \text{if } \|\theta\|_2 \neq b \\ b^{-d+1}, & \text{otherwise} \end{cases}.$$

The above optimization problem can be further rewritten as

$$\operatorname*{argmin}_{\hat{\theta} \in \mathcal{D}_G} R(\hat{\theta}, \Pi_t),$$

where $R(\hat{\theta}, \Pi_t) \stackrel{\text{def}}{=} \mathbb{E}_{\theta \sim \Pi_t} \left[R(\hat{\theta}, \theta) \right]$, and Π_t is the distribution of a random variable θ which is generated by first sampling b from P_t and then generating a sample from U_b. Note that Π_t is a spherically symmetric distribution. From Lemma 5.2, we know that the Bayes estimator corresponding to any invariant prior is an invariant estimator. So the minimization over $\mathcal{D}_G$ in the above optimization problem can be replaced with minimization over the set of all estimators $\mathcal{D}$. This leads us to the following equivalent optimization problem

$$\operatorname*{argmin}_{\hat{\theta} \in \mathcal{D}} R(\hat{\theta}, \Pi_t).$$

Let $\hat{\theta}_t$ be the minimizer of this equivalent problem. We now obtain an expression for $\hat{\theta}_t(X)$ in terms of modified Bessel functions. Let $\Pi_t(\cdot|X)$ be the posterior distribution of θ given the data X and let $p(X; \theta)$ be the probability density function for distribution P_θ. Since the risk is measured with respect to ℓ_2^2 metric, the Bayes estimator $\hat{\theta}_t(X)$ is given by the

posterior mean

$$\begin{aligned}
\hat{\theta}_t(X) &= \mathbb{E}_{\theta\sim\Pi_t(\cdot|X)}\left[\theta\right] \\
&= \frac{\mathbb{E}_{\theta\sim\Pi_t}\left[\theta p(X;\theta)\right]}{\mathbb{E}_{\theta\sim\Pi_t}\left[p(X;\theta)\right]} \\
&= \frac{\mathbb{E}_{b\sim P_t}\left[\int \theta u_b(\theta) p(X;\theta) d\theta\right]}{\mathbb{E}_{b\sim P_t}\left[\int u_b(\theta) p(X;\theta) d\theta\right]} \quad \text{(definition of } \Pi_t\text{)} \\
&= \frac{\mathbb{E}_{b\sim P_t}\left[b^{-d+1}\int_{\|\theta\|_2=b} \theta p(X;\theta) d\theta\right]}{\mathbb{E}_{b\sim P_t}\left[b^{-d+1}\int_{\|\theta\|_2=b} p(X;\theta) d\theta\right]} \quad \text{(since } U_b \text{ is uniform on sphere)} \\
&= \frac{\mathbb{E}_{b\sim P_t}\left[b^{-d+1}e^{-b^2/2}\int_{\|\theta\|_2=b} \theta e^{\langle X,\theta\rangle} d\theta\right]}{\mathbb{E}_{b\sim P_t}\left[b^{-d+1}e^{-b^2/2}\int_{\|\theta\|_2=b} e^{\langle X,\theta\rangle} d\theta\right]} \\
&= \frac{\mathbb{E}_{b\sim P_t}\left[b^{2}e^{-b^2/2}\int_{\|\theta\|_2=1} \theta e^{b\langle X,\theta\rangle} d\theta\right]}{\mathbb{E}_{b\sim P_t}\left[be^{-b^2/2}\int_{\|\theta\|_2=1} e^{b\langle X,\theta\rangle} d\theta\right]} \quad \text{(change of variables).}
\end{aligned}$$

We now obtain a closed-form expression for the terms $\int_{\|\theta\|_2=1} \theta e^{b\langle X,\theta\rangle} d\theta$ and $\int_{\|\theta\|_2=1} e^{b\langle X,\theta\rangle} d\theta$ appearing in the RHS of the above equation. We do this by relating them to the mean and normalization constant of Von Mises-Fisher (vMF) distribution, which is a probability distribution on the unit sphere centered at origin in $\mathbb{R}^d$. This distribution is usually studied in directional statistics [79]. The probability density function of a random unit vector $Z \in \mathbb{R}^d$ distributed according to vMF distribution is given by

$$p(Z;\mu,\kappa) = \frac{\kappa^{d/2-1}}{(2\pi)^{d/2} I_{d/2-1}(\kappa)} \exp(\kappa \langle \mu, Z\rangle),$$

where $\kappa \geq 0, \|\mu\|_2 = 1$, I_ν is the modified Bessel function of the first kind of order ν. Using the fact that a probability density function integrates to 1, we get the following closed-form expression for $\int_{\|\theta\|_2=1} e^{b\langle X,\theta\rangle} d\theta$

$$\int_{\|\theta\|_2=1} e^{b\langle X,\theta\rangle} d\theta = \frac{(2\pi)^{d/2} I_{d/2-1}(b\|X\|_2)}{(b\|X\|_2)^{d/2-1}}. \tag{5.23}$$

To get a closed-form expression for $\int_{\|\theta\|_2=1} \theta e^{b\langle X,\theta\rangle} d\theta$, we relate it to mean of vMF distribution. We have the following expression for the mean of a random vector distributed according to vMF distribution [6]

$$\int_{\|Z\|=1} Z p(Z;\mu,\kappa) dZ = \frac{I_{d/2}(\kappa)}{I_{d/2-1}(\kappa)} \mu.$$

Using the above equality, we get the following expression for $\int_{\|\theta\|_2=1} \theta e^{b\langle X,\theta\rangle} d\theta$

$$\int_{\|\theta\|_2=1} \theta e^{b\langle X,\theta\rangle} d\theta = \frac{(2\pi)^{d/2} I_{d/2}(b\|X\|_2)}{(b\|X\|_2)^{d/2-1}} \frac{X}{\|X\|_2}. \tag{5.24}$$

Substituting Equations (5.23), (5.24) in the expression for $\hat{\theta}_t(X)$ obtained above, we get an expression for $\hat{\theta}_t(X)$ which involves the modified Bessel function I_ν and integrals over variable b. We note that I_ν can be computed to very high accuracy and there exist accurate implementations of I_ν in a number of programming languages. So in our analysis of the approximation error of Algorithm 5.3, we assume the error from the computation of I_ν is 0.

5.8.7.2 Proof of Theorem 5.9

Before we present the proof of the Theorem we present useful intermediate results which we require in our proof.

Intermediate Results

Lemma 5.3 (Lipschitz Continuity)**.** *Consider the problem of finite Gaussian sequence model. Let $\Theta = \{\theta : \theta \in \mathbb{R}^d, \|\theta\|_2 \leq B\}$ be the ball of radius B centered at origin in $\mathbb{R}^d$. Let $\hat{\theta}$ be any estimator which maps X to an element in Θ. Then the risk $R(\hat{\theta}, \theta) = \mathbb{E}_{X\sim\mathcal{N}(\theta,I)}\left[\|\hat{\theta}(X) - \theta\|_2^2\right]$ is Lipschitz continuous in its second argument w.r.t ℓ_2 norm over the domain Θ, with Lipschitz constant $4(B + \sqrt{d}B^2)$. Moreover, $R(\hat{\theta}, b\mathbf{e}_1) = \mathbb{E}_{X\sim\mathcal{N}(\theta,I)}\left[\|\hat{\theta}(X) - b\mathbf{e}_1\|_2^2\right]$ is Lipschitz continuous in b over the domain $[0, B]$, with Lipschitz constant $4(B + B^2)$.*

Proof. Let $R_{\hat{\theta}}(\theta) = R(\hat{\theta}, \theta)$. The gradient of $R_{\hat{\theta}}(\theta)$ with respect to θ is given by

$$\nabla_\theta R_{\hat{\theta}}(\theta) = \mathbb{E}_{X\sim\mathcal{N}(\theta,I)}\left[2(\theta - \hat{\theta}(X)) + (X - \theta)\|\hat{\theta}(X) - \theta\|_2^2\right].$$

The norm of $\nabla_\theta R_{\hat{\theta}}(\theta)$ can be upper bounded as follows

$$\begin{aligned}
\|\nabla_\theta R_{\hat{\theta}}(\theta)\|_2 &\leq \left|\left|\mathbb{E}_{X\sim\mathcal{N}(\theta,I)}\left[2(\theta - \hat{\theta}(X))\right]\right|\right|_2 + \left|\left|\mathbb{E}_{X\sim\mathcal{N}(\theta,I)}\left[(X - \theta)\|\hat{\theta}(X) - \theta\|_2^2\right]\right|\right|_2 \\
&\overset{(a)}{\leq} 4B + \mathbb{E}_{\mathbf{x}\sim\mathcal{N}(\theta,I)}\left[\|X - \theta\|_2\|\hat{\theta}(X) - \theta\|_2^2\right] \\
&\overset{(b)}{\leq} 4B + 4B^2\mathbb{E}_{X\sim\mathcal{N}(\theta,I)}\left[\|X - \theta\|_2\right] \\
&\leq 4B + 4\sqrt{d}B^2,
\end{aligned}$$

where the first term in (a) follows from the fact that $\theta, \hat{\theta}(X) \in \Theta$ and the second term follows from Jensen's inequality. This shows that $R_{\hat{\theta}}(\theta)$ is Lipschitz continuous over Θ. This finishes the first part of the proof. To show that $R(\hat{\theta}, b\mathbf{e}_1)$ is Lipschitz continuous in

b, we use similar arguments. Let $R_{\hat{\theta}}(b) = R(\hat{\theta}, b\mathbf{e}_1)$. Then

$$\begin{aligned}
\left|R'_{\hat{\theta}}(b)\right| &= \left| \left\langle \mathbf{e}_1, \nabla_\theta R_{\hat{\theta}}(\theta)\Big|_{\theta=b\mathbf{e}_1} \right\rangle \right| \\
&\overset{(a)}{\leq} \left| \mathbb{E}_{X\sim\mathcal{N}(b\mathbf{e}_1,I)} \left[2(b - [\hat{\theta}(X)]_1)\right] \right| + \left|\left| \mathbb{E}_{X\sim\mathcal{N}(b\mathbf{e}_1,I)} \left[(X_1 - b)\|\hat{\theta}(X) - b\mathbf{e}_1\|_2^2\right] \right|\right|_2 \\
&\leq 4B + \mathbb{E}_{\mathbf{x}\sim\mathcal{N}(b\mathbf{e}_1,I)} \left[|X_1 - b|\|\hat{\theta}(X) - b\mathbf{e}_1\|_2^2\right] \\
&\leq 4B + 4B^2\mathbb{E}_{X\sim\mathcal{N}(b\mathbf{e}_1,I)} \left[|X_1 - b|\right] \\
&\leq 4B + 4B^2,
\end{aligned}$$

where (a) follows from the expression for $\nabla_\theta R_{\hat{\theta}}(\theta)$ obtained above. □

Lemma 5.4 (Approximation of risk)**.** *Consider the setting of Lemma 5.3. Let $\hat{\theta}$ be any estimator which maps X to an element in Θ. Let $\{X_i\}_{i=1}^N$ be N i.i.d samples from $\mathcal{N}(\theta, I)$. Then with probability at least $1 - \delta$*

$$\left| \frac{1}{N} \sum_{i=1}^{N} \|\hat{\theta}(X_i) - \theta\|_2^2 - R_{\hat{\theta}}(\theta) \right| \leq 4B^2 \sqrt{\frac{\log \frac{1}{\delta}}{N}}.$$

Proof. The proof of the Lemma relies on concentration properties of sub-Gaussian random variables. Let $Z(X) = \|\hat{\theta}(X) - \theta\|^2$. Note that $R_{\hat{\theta}}(\theta) = \mathbb{E}_{X\sim\mathcal{N}(\theta,I)}[Z(X)]$. Since $Z(X)$ is bounded by $4B^2$, it is a sub-Gaussian random variable. Using Hoeffding bound we get

$$\left| \frac{1}{N} \sum_{i=1}^{N} Z(X_i) - \mathbb{E}\left[Z(X)\right] \right| \leq 4B^2 \sqrt{\frac{\log \frac{1}{\delta}}{N}}, \quad \text{w.p} \geq 1 - \delta.$$

□

Main Argument The proof relies on Corollary 5.2 to show that the averaged estimator $\hat{\theta}_{AVG}$ is approximately minimax and $\hat{P}_{\text{LFP}}$ is approximately least favorable. Here is a rough sketch of the proof. We first apply the corollaries on the following reduced statistical game that we are aiming to solve

$$\inf_{\hat{\theta}\in\mathcal{D}_G} \sup_{b\in[0,B]} R(\hat{\theta}, b\mathbf{e}_1).$$

To apply these corollaries, we need the risk $R(\hat{\theta}, b\mathbf{e}_1)$ to be Lipscthiz continuous in b. This holds for us because of Lemma 5.3. Next, we convert the guarantees for the reduced statistical game to the orginial statistical game to show that we learn a minimax estimator and LFP for finite Gaussian sequence model.

To use Corollary 5.2, we first need to bound α, β, α', the approximation errors of the optimization subroutines described in Algorithms 5.2, 5.3. A major part of the proof involves bounding these quantities.

Approximation error of Algorithm 5.2. There are two causes for error in the optimization oracle described in Algorithm 5.2: (a) grid search and (b) approximate computation of risk $R(\hat{\theta}, b\mathbf{e}_1)$. We now bound the error due to both (a) and (b). From Lemma 5.4 we know that for any estimator $\hat{\theta}_i$ and grid point b_j, the following holds with probability at least $1-\delta$

$$\Big|\frac{1}{N_1}\sum_{k=1}^{N_1}\|\hat{\theta}_i(X_k) - b_j\mathbf{e}_1\|_2^2 - R(\hat{\theta}_i, b_j\mathbf{e}_1)\Big| \le 4B^2\sqrt{\frac{\log\frac{1}{\delta}}{N_1}}.$$

Taking a union bound over all estimators $\{\hat{\theta}_i\}_{i=1}^T$ and grid points $\{b_j\}_{j=1}^{B/w}$, we can show that with probability at least $1-\delta$, the following holds for all $i \in [T], j \in [B/w]$

$$\Big|\frac{1}{N_1}\sum_{k=1}^{N_1}\|\hat{\theta}_i(X_k) - b_j\mathbf{e}_1\|_2^2 - R(\hat{\theta}_i, b_j\mathbf{e}_1)\Big| \le 4B^2\sqrt{\frac{\log\frac{BT}{w\delta}}{N_1}}. \tag{5.25}$$

Let $f_{t,\sigma}(b)$ be the actual objective we would like to optimize in iteration t of Algorithm 5.1, which is given by

$$f_{t,\sigma}(b) = \sum_{i=1}^{t-1} R(\hat{\theta}_i, b\mathbf{e}_1) + \sigma b.$$

Let $\hat{f}_{t,\sigma}(b)$ be the approximate objective we are optimizing by replacing $R(\hat{\theta}_i, b\mathbf{e}_1)$ with its approximate estimate. Let b_t^* be a maximizer of $f_{t,\sigma}(b)$ and $b_{t,\text{approx}}^*$ be the maximizer of $\hat{f}_{t,\sigma}(b)$ (which is also the output of Algorithm 5.2). Finally, let $b_{t,\text{NN}}^*$ be the point on the grid which is closest to b_t^*. Using Lemma 5.3 we first show that $f_{t,\sigma}(b)$ is Lipschitz continuous in b. The derivative of $f_{t,\sigma}(b)$ with respect to b is given by

$$f'_{t,\sigma}(b) = \sum_{i=1}^{t-1}\left\langle \mathbf{e}_1, \nabla_\theta R(\hat{\theta}_i, \theta)\Big|_{\theta=b\mathbf{e}_1}\right\rangle + \sigma$$

Using Lemma 5.3, the magnitude of $f'_{t,\sigma}(b)$ can be upper bounded as

$$|f'_{t,\sigma}(b)| \le 4(t-1)(B+B^2) + \sigma.$$

This shows that $f_{t,\sigma}(b)$ is Lipschitz continuous in b. We now bound $f_{t,\sigma}(b_t^*) - f_{t,\sigma}(b_{t,\text{approx}}^*)$, the approximation error of the optimization oracle

$$\begin{aligned}
f_{t,\sigma}(b_t^*) &\overset{(a)}{\le} f_{t,\sigma}(b_{t,\text{NN}}^*) + \big(4t(B+B^2)+\sigma\big)\,w \\
&\overset{(b)}{\le} \hat{f}_{t,\sigma}(b_{t,\text{NN}}^*) + 4tB^2\sqrt{\frac{\log\frac{BT}{w\delta}}{N_1}} + \big(4t(B+B^2)+\sigma\big)\,w \\
&\overset{(c)}{\le} \hat{f}_{t,\sigma}(b_{t,\text{approx}}^*) + 4tB^2\sqrt{\frac{\log\frac{BT}{w\delta}}{N_1}} + \big(4t(B+B^2)+\sigma\big)\,w \\
&\overset{(d)}{\le} f_{t,\sigma}(b_{t,\text{approx}}^*) + 8tB^2\sqrt{\frac{\log\frac{BT}{w\delta}}{N_1}} + \big(4t(B+B^2)+\sigma\big)\,w,
\end{aligned}$$

where (a) follows from Lipschitz property of the loss function and $(b), (d)$ follow from Equation (5.25) and hold with probability at least $1-\delta$ and (c) follows from the optimality

of $b^*_{t,\text{approx}}$. This shows that Algorithm 5.2 is a $\left(O\left(TB^2\sqrt{\frac{\log\frac{BT}{w\delta}}{N_1}}+TB(1+B)w\right), w\right)$-approximate maximization oracle; that is

$$\alpha = O\left(TB^2\sqrt{\frac{\log\frac{BT}{w\delta}}{N_1}}+TB(1+B)w\right), \quad \beta = w.$$

Approximation error of Algorithm 5.3. There are two sources of approximation error in Algorithm 5.3: (a) computation of modified Bessel functions I_ν, and (b) approximation of P_t with its samples. In this analysis we assume that I_ν can be computed to very high accuracy. This is a reasonable assumption because many programming languages have accurate and efficient implementations of I_ν. So the main focus here is on bounding the error from approximation of P_t.

First, note that since we are using grid search to optimize the maximization problem, the true distribution P_t for which we are supposed to compute the Bayes estimator is a discrete distribution supported on grid points $\{b_1, \dots b_{B/w}\}$. Algorithm 5.3 does not compute the Bayes estimator for P_t. Instead, we generate samples from P_t and use them as a proxy for P_t. Let $\hat{P}_t$ be the empirical distribution obtained by sampling N_2 points from P_t. Let $p_{t,j}$ be the probability mass on grid point b_j. Using Bernstein inequality we can show that the following holds with probability at least $1-\delta$

$$\forall j \in [B/w] \quad |\hat{p}_{t,j} - p_{t,j}| \le \sqrt{p_{t,j}\frac{\log\frac{B}{w\delta}}{N_2}}. \tag{5.26}$$

Define estimators $\hat{\theta}'_t, \hat{\theta}_t$ as

$$\hat{\theta}'_t \leftarrow \operatorname*{argmin}_{\hat{\theta}\in\mathcal{D}_G} \mathbb{E}_{b\sim P_t}\left[R(\hat{\theta}, b\mathbf{e}_1)\right], \quad \hat{\theta}_t \leftarrow \operatorname*{argmin}_{\hat{\theta}\in\mathcal{D}_G} \mathbb{E}_{b\sim \hat{P}_t}\left[R(\hat{\theta}, b\mathbf{e}_1)\right].$$

$\hat{\theta}'_t$ is what we ideally want to compute. $\hat{\theta}_t$ is what we end up computing using Algorithm 5.3. We now show that $\hat{\theta}_t$ is an approximate minimizer of the left hand side optimization problem above. To this end, we try to bound the following quantity

$$\mathbb{E}_{b\sim P_t}\left[R(\hat{\theta}_t, b\mathbf{e}_1) - R(\hat{\theta}'_t, b\mathbf{e}_1)\right].$$

Let $f_t(\hat{\theta}) = \mathbb{E}_{b\sim P_t}\left[R(\hat{\theta}, b\mathbf{e}_1)\right]$ and $\hat{f}_t(\hat{\theta}) = \mathbb{E}_{b\sim \hat{P}_t}\left[R(\hat{\theta}, b\mathbf{e}_1)\right]$. We would like to bound the quantity $f_t(\hat{\theta}_t) - f_t(\hat{\theta}'_t)$. Consider the following

$$\begin{aligned} f_t(\hat{\theta}_t) &\overset{(a)}{\le} \hat{f}_t(\hat{\theta}_t) + \frac{4B^3}{w}\sqrt{\frac{\log\frac{B}{w\delta}}{N_2}} \\ &\overset{(b)}{\le} \hat{f}_t(\hat{\theta}'_t) + \frac{4B^3}{w}\sqrt{\frac{\log\frac{B}{w\delta}}{N_2}} \\ &\overset{(c)}{\le} f_t(\hat{\theta}'_t) + \frac{8B^3}{w}\sqrt{\frac{\log\frac{B}{w\delta}}{N_2}}, \end{aligned}$$

where (a) follows from Equation (5.26) and the fact that the risk $R(\hat{\theta}, \theta)$ of any estimator is bounded by $4B^2$, (b) follows since $\hat{\theta}_t$ is a minimizer of $\hat{f}_t$ and (c) follows from Equation (5.26). This shows that with probability at least $1-\delta$, Algorithm 5.3 is an $O\left(\frac{B^3}{w}\sqrt{\frac{\log \frac{B}{w\delta}}{N_2}}\right)$-approximate optimization oracle; that is,

$$\alpha' = O\left(\frac{B^3}{w}\sqrt{\frac{\log \frac{B}{w\delta}}{N_2}}\right).$$

Minimax Estimator. We are now ready to show that $\hat{\theta}_{AVG}$ is an approximate minimax estimator. Instantiating Corollary 5.2 for the reduced statistical game gives us the following bound, which holds with probability at least $1-\delta$

$$\sup_{b\in[0,B]} R(\hat{\theta}_{AVG}, b\mathbf{e}_1) \leq \inf_{\hat{\theta}\in\mathcal{D}_G} \sup_{b\in[0,B]} R(\hat{\theta}, b\mathbf{e}_1) + \tilde{O}\left(\frac{B^2(B+1)}{\sqrt{T}} + \alpha + \alpha' + \beta B(B+1)\sqrt{T}\right),$$

where we used the fact that the risk $R(\hat{\theta}, b\mathbf{e}_1)$ is $4B(B+1)$-Lipschitz continuous w.r.t b. The $\tilde{O}$ notation in the above inequality hides logarithmic factors. Plugging in the values of α, α', β in the above equation gives us

$$\sup_{b\in[0,B]} R(\hat{\theta}_{AVG}, b\mathbf{e}_1) \leq \inf_{\hat{\theta}\in\mathcal{D}_G} \sup_{b\in[0,B]} R(\hat{\theta}, b\mathbf{e}_1) + \tilde{O}\left(\frac{B^2(B+1)}{\sqrt{T}}\right).$$

We now convert this bound to a bound on the original statistical game. From Theorem 5.3 we know that $\inf_{\hat{\theta}\in\mathcal{D}_G} \sup_{b\in[0,B]} R(\hat{\theta}, b\mathbf{e}_1) = \inf_{\hat{\theta}\in\mathcal{D}} \sup_{\theta\in\Theta} R(\hat{\theta}, \theta) = R^*$. Since the estimator $\hat{\theta}_{AVG}$ is invariant to transformations of orthogonal group, we have $R(\hat{\theta}_{AVG}, \theta) = R(\hat{\theta}_{AVG}, \|\theta\|_2\mathbf{e}_1)$ for any $\theta \in \Theta$. Using these two results in the above inequality, we get

$$\sup_{\theta\in\Theta} R(\hat{\theta}_{AVG}, \theta) = \sup_{b\in[0,B]} R(\hat{\theta}_{AVG}, b\mathbf{e}_1) \leq R^* + \tilde{O}\left(\frac{B^2(B+1)}{\sqrt{T}}\right).$$

This shows that the worst-case risk of $\hat{\theta}_{AVG}$ is close to the minimax risk R^*. This finishes the first part of the proof.

LFP. To prove the second part, we rely on Corollary 5.2. Instantiating it for the reduced statistical game gives us

$$\inf_{\hat{\theta}\in\mathcal{D}_G} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{b\sim P_t}\left[R(\hat{\theta}, b\mathbf{e}_1)\right] \geq R^* - \tilde{O}\left(\frac{B^2(B+1)}{\sqrt{T}} + \alpha + \alpha' + \beta B(B+1)\sqrt{T}\right).$$

Plugging in the values of α, α', β in the above equation gives us

$$\inf_{\hat{\theta}\in\mathcal{D}_G} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{b\sim P_t}\left[R(\hat{\theta}, b\mathbf{e}_1)\right] \geq R^* - \tilde{O}\left(\frac{B^2(B+1)}{\sqrt{T}}\right).$$

From Equation (5.26) we know that P_t is close to $\hat{P}_t$ with high probability. Using this, we can replace P_t in the above bound with $\hat{P}_t$ and obtain the following bound, which holds with probability at least $1 - \delta$

$$\inf_{\hat{\theta} \in \mathcal{D}_G} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{b \sim \hat{P}_t} \left[R(\hat{\theta}, b\mathbf{e}_1) \right] \geq R^* - \tilde{O}\left(\frac{B^2(B+1)}{\sqrt{T}} \right). \tag{5.27}$$

In the rest of the proof, we show that $\inf_{\hat{\theta} \in \mathcal{D}_G} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{b \sim \hat{P}_t} \left[R(\hat{\theta}, b\mathbf{e}_1) \right] = \inf_{\hat{\theta}} R(\hat{\theta}, \hat{P}_{\text{LFP}})$. Recall, the density function of $\hat{P}_{\text{LFP}}$ is given by: $\hat{p}_{\text{LFP}}(\theta) \propto \|\theta\|_2^{1-d} \hat{P}_{\text{AVG}}(\|\theta\|_2)$, where $\hat{P}_{\text{AVG}}(\|\theta\|_2)$ is the probability mass placed by $\hat{P}_{\text{AVG}}$ at $\|\theta\|_2$. This distribution is equivalent to the distribution of a random variable which is generated by first sampling b from $\hat{P}_t$ and then sampling θ from the uniform distribution on $(d-1)$ dimensional sphere of radius b, centered at origin in $\mathbb{R}^d$. Using this equivalence, we can equivalently rewrite $R(\hat{\theta}, \hat{P}_{\text{LFP}})$ for any estimator $\hat{\theta}$ as

$$R(\hat{\theta}, \hat{P}_{\text{LFP}}) = \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{b \sim \hat{P}_t} \left[\mathbb{E}_{\theta \sim U} \left[R(\hat{\theta}, b\theta) \right] \right],$$

where U is the uniform distribution on the $(d-1)$ dimensional unit sphere centered at origin, in $\mathbb{R}^d$. Next, from Lemma 5.2, we know that the Bayes estimator corresponding to any invariant prior is an invariant estimator. Since $\hat{P}_{\text{LFP}}$ is an invariant distribution, we have

$$\inf_{\hat{\theta} \in \mathcal{D}} R(\hat{\theta}, \hat{P}_{\text{LFP}}) = \inf_{\hat{\theta} \in \mathcal{D}_G} R(\hat{\theta}, \hat{P}_{\text{LFP}}) = \inf_{\hat{\theta} \in \mathcal{D}_G} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{b \sim \hat{P}_t} \left[\mathbb{E}_{\theta \sim U} \left[R(\hat{\theta}, b\theta) \right] \right].$$

From Lemma 5.1 we know that for any invariant estimator $\hat{\theta}$, we have $R(\hat{\theta}, \theta_1) = R(\hat{\theta}, \theta_2)$, whenever $\theta_1 \sim \theta_2$. Using this result in the above equation gives us

$$\inf_{\hat{\theta} \in \mathcal{D}} R(\hat{\theta}, \hat{P}_{\text{LFP}}) = \inf_{\hat{\theta} \in \mathcal{D}_G} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{b \sim \hat{P}_t} \left[R(\hat{\theta}, b\mathbf{e}_1) \right].$$

Combining the above result with Equation (5.27) shows that $\hat{P}_{\text{LFP}}$ is approximately least favorable.

5.8.8 Loss on few co-ordinates

In this section, we present the optimization oracles for the problem of finite Gaussian sequence model, when the loss is evaluated on a few co-ordinates. Recall, in Theorem 5.5 we showed that the original min-max statistical game can be reduced to the following simpler problem

$$\inf_{\hat{\theta} \in \mathcal{M}_{\mathcal{D},G}} \sup_{b : b[1]^2 + b[2]^2 \leq B^2} R(\hat{\theta}, [b[1]\mathbf{e}_{1,k}, b[2]\mathbf{e}_{1,d-k}]), \tag{5.28}$$

where $b[j]$ represents the j^{th} co-ordinate of b. We now provide efficient implementations of the optimization oracles required by Algorithm 5.1 for finding a Nash equilibrium of this

game. The optimization problems corresponding to the two optimization oracles are as follows

$$\hat{\theta}_t \leftarrow \operatorname*{argmin}_{\hat{\theta} \in \mathcal{D}_G} \mathbb{E}_{b \sim P_t} \left[R(\hat{\theta}, [b[1]\mathbf{e}_{1,k}, b[2]\mathbf{e}_{1,d-k}]) \right],$$

$$b_t(\sigma) \leftarrow \operatorname*{argmax}_{b: b[1]^2 + b[2]^2 \leq B^2} \sum_{i=1}^{t-1} R(\hat{\theta}_i, [b[1]\mathbf{e}_{1,k}, b[2]\mathbf{e}_{1,d-k}]) + \langle \sigma, b \rangle,$$

where $\mathcal{D}_G$ is the set of deterministic invariant estimators and P_t is the distribution of random variable $b_t(\sigma)$. The maximization oracle can be efficiently implemented via a grid search over $\{b : b[1]^2 + b[2]^2 \leq B^2\}$ (see Algorithm 5.7). The minimization oracle can also be efficiently implemented. The minimizer has a closed form expression which depends on P_t and modified Bessel functions (see Algorithm 5.8).

Algorithm 5.7 Maximization Oracle

1: **Input:** Number of coordinates to evaluate loss on k, estimators $\{\hat{\theta}_i\}_{i=1}^{t-1}$, perturbation σ, grid width w, number of samples for computation of expected risk $R(\hat{\theta}, \theta)$: N_1
2: Let $\{b_1, b_2 \dots b_{N(w)}\}$ be the w-covering of $\{b : b[1]^2 + b[2]^2 \leq B^2\}$
3: **for** $j = 1 \dots N(w)$ **do**
4: **for** $i = 1 \dots t-1$ **do**
5: Generate N_1 independent samples $\{X_l\}_{l=1}^{N_1}$ from the following distribution

$$\mathcal{N}([b_j[1]\mathbf{e}_{1,k}, b_j[2]\mathbf{e}_{1,d-k}], I)$$

6: Estimate $R(\hat{\theta}_i, [b_j[1]\mathbf{e}_{1,k}, b_j[2]\mathbf{e}_{1,d-k}])$ as

$$\frac{1}{N_1} \sum_{l=1}^{N_1} \|\hat{\theta}_i(X_l)[1:k] - b_j[1]\mathbf{e}_{1,k}\|_2^2.$$

7: Evaluate the objective at b_j using the above estimates
8: **Output:** b_j which maximizes the objective

Algorithm 5.8 Minimization Oracle

1: **Input:** Samples $\{b_i\}_{i=1}^{N_2}$ generated from distribution P_t, number of coordinates to evaluate loss on k.
2: For any X, compute $\hat{\theta}_t(X)$ as

$$\left(\frac{\sum_{i=1}^{N_2} w_i b_i[1] A_k(b_i[1] \|X[1:k]\|_2)}{\sum_{i=1}^{N_2} w_i} \right) \frac{X[1:k]}{\|X[1:k]\|_2},$$

where $A_k(\gamma) = \dfrac{I_{k/2}(\gamma)}{I_{k/2-1}(\gamma)}$,

$$w_i = b_i[1]^{2-\frac{k}{2}} b_i[2]^{2-\frac{d-k}{2}} e^{-\frac{\|b\|^2}{2}} I_{k/2-1}(b_i[1]\|X[1:k]\|_2) I_{(d-k)/2-1}(b_i[2]\|X[k+1:d]\|_2),$$

and I_ν is the modified Bessel function of the first kind of order ν.

5.8.9 Linear Regression

5.8.9.1 Proof of Proposition 5.3

In this section we derive a closed-form expression for the minimizer $\hat{\theta}_t$ of the following objective

$$\operatorname*{argmin}_{\hat{\theta} \in \mathcal{D}_G} \mathbb{E}_{b \sim P_t} \left[R(\hat{\theta}, b\mathbf{e}_1) \right].$$

Using the same arguments as in proof of Proposition 5.2, we can show that the above optimization problem can be rewritten as the following equivalent optimization problem over the set of all deterministic estimators

$$\operatorname*{argmin}_{\hat{\theta} \in \mathcal{D}} \mathbb{E}_{\theta \sim \Pi_t} \left[R(\hat{\theta}, \theta) \right],$$

where Π_t is the distribution of a random variable θ which is generated by first sampling a b from P_t and then drawing a random sample from U_b, the uniform distribution on a spherical shell of radius b. The density function of U_b is given by

$$u_b(\theta) \propto \begin{cases} 0, & \text{if } \|\theta\|_2 \neq b \\ b^{-d+1}, & \text{otherwise} \end{cases}.$$

Since the risk is measured with respect to ℓ_2^2 metric, the minimizer $\hat{\theta}_t(D_n)$ is given by the posterior mean

$$\begin{aligned}
\hat{\theta}_t(D_n) &= \mathbb{E}_{\theta \sim \Pi_t(\cdot|D_n)} [\theta] \\
&= \frac{\mathbb{E}_{\theta \sim \Pi_t} [\theta p(D_n; \theta)]}{\mathbb{E}_{\theta \sim \Pi_t} [p(D_n; \theta)]} \\
&= \frac{\mathbb{E}_{b \sim P_t} \left[\int \theta u_b(\theta) p(D_n; \theta) d\theta \right]}{\mathbb{E}_{b \sim P_t} \left[\int u_b(\theta) p(D_n; \theta) d\theta \right]} \\
&= \frac{\mathbb{E}_{b \sim P_t} \left[b^{-d+1} \int_{\|\theta\|_2 = b} \theta p(D_n; \theta) d\theta \right]}{\mathbb{E}_{b \sim P_t} \left[b^{-d+1} \int_{\|\theta\|_2 = b} p(D_n; \theta) d\theta \right]} \\
&= \frac{\mathbb{E}_{b \sim P_t} \left[b^{-d+1} \int_{\|\theta\|_2 = b} \theta e^{-\frac{\|\mathbf{Y} - \mathbf{X}\theta\|_2^2}{2}} d\theta \right]}{\mathbb{E}_{b \sim P_t} \left[b^{-d+1} \int_{\|\theta\|_2 = b} e^{-\frac{\|\mathbf{Y} - \mathbf{X}\theta\|_2^2}{2}} d\theta \right]} \\
&= \frac{\mathbb{E}_{b \sim P_t} \left[b^2 \int_{\|\theta\|_2 = 1} \theta e^{-\frac{b^2 \|\mathbf{X}\theta\|_2^2 - 2b\langle \theta, \mathbf{X}^T \mathbf{Y} \rangle}{2}} d\theta \right]}{\mathbb{E}_{b \sim P_t} \left[b \int_{\|\theta\|_2 = 1} e^{-\frac{b^2 \|\mathbf{X}\theta\|_2^2 - 2b\langle \theta, \mathbf{X}^T \mathbf{Y} \rangle}{2}} d\theta \right]} \qquad \text{(change of variables).}
\end{aligned}$$

We now relate the terms appearing in the above expression to the mean and normalization constant of Fisher-Bingham (FB) distribution. As stated in Section 5.4, the probability density function of a random unit vector $Z \in \mathbb{R}^d$ distributed according to FB distribution is given by

$$p(Z; A, \gamma) = C(A, \gamma)^{-1} \exp\left(-Z^T A Z + \langle \gamma, Z \rangle \right),$$

where $Z \in \mathbb{S}^{d-1}$, and $\gamma \in \mathbb{R}^d$, $A \in \mathbb{R}^{d \times d}$ are the parameters of the distribution with A being positive semi-definite and $C(A, \gamma)$ is the normalization constant which is given by

$$C(A, \gamma) = \int_{\|Z\|_2=1} \exp\left(-Z^T A Z + \langle \gamma, Z \rangle\right) dZ.$$

The mean of Z is given by

$$\int_{\|Z\|_2=1} Z p(Z; A, \gamma) dZ = C(A, \gamma)^{-1} \int_{\|Z\|_2=1} Z \exp\left(-Z^T A Z + \langle \gamma, Z \rangle\right) dZ = C(A, \gamma)^{-1} \frac{\partial}{\partial \gamma} C(A, \gamma).$$

Using these in the previously derived expression for $\hat{\theta}(D_n)$ gives us the required result.

5.8.9.2 Mean and normalization constant of Fisher-Bingham distribution

In this section, we present our technique for computation of $C(A, \gamma)$. Once we have an accurate technique for its computation, computing $\frac{\partial}{\partial \gamma} C(A, \gamma)$ should be straight forward as one can rely on efficient numerical differentiation techniques for its computation. Recall, to implement Algorithm 5.5 we need to compute $C\left(2^{-1} b^2 \mathbf{X}^T \mathbf{X}, b \mathbf{X}^T \mathbf{Y}\right)$. Let $\hat{\Sigma} = \frac{1}{n} \mathbf{X}^T \mathbf{X}$ and let $U \Lambda U^T$ be its eigen decomposition. Then it is easy to see that $C\left(2^{-1} b^2 \mathbf{X}^T \mathbf{X}, b \mathbf{X}^T \mathbf{Y}\right)$ can be rewritten as

$$C\left(2^{-1} b^2 \mathbf{X}^T \mathbf{X}, b \mathbf{X}^T \mathbf{Y}\right) = C(2^{-1} n b^2 \Lambda, b U^T \mathbf{X}^T \mathbf{Y}).$$

So it suffices to compute $C(A, \gamma)$ for some positive semi-definite, diagonal matrix A and vector γ. Let a_i be the i^{th} diagonal entry of A and let γ_i be the i^{th} element of γ. Kume and Wood [68] derive the following expression for $C(A, \gamma)$

$$C(A, \gamma) = (2\pi)^{d/2} \left(\prod_{i=1}^{d} a_i^{-1/2}\right) \exp\left(\frac{1}{4} \sum_{i=1}^{d} \frac{\gamma_i^2}{a_i}\right) f_{A,\gamma}(1),$$

where $f_{A,\gamma}$ is the probability density of a non-central chi-squared random variable $\sum_{i=1}^{d} z_i^2$ with $z_i \sim \mathcal{N}(\frac{\gamma_i}{2a_i}, \frac{1}{2a_i})$. There are number of efficient techniques for computation of $f_{A,\gamma}(1)$ [51, 68]. We first present the technique of Imhof [51] for exact computation of $f_{A,\gamma}(1)$. Imhof [51] showed that $f_{A,\gamma}(1)$ can be written as the following integral

$$f_{A,\gamma}(1) = \pi^{-1} \int_0^{\infty} [\rho(u)]^{-1} \cos \zeta(u) du,$$

where $\rho : \mathbb{R} \to \mathbb{R}$ and $\zeta : \mathbb{R} \to \mathbb{R}$ are defined as

$$\zeta(u) = \frac{1}{2} \sum_{i=1}^{d} \left(\tan^{-1}\left(\frac{u}{2a_i}\right) + \frac{\gamma_i^2}{8a_i^3} \left(1 + \frac{u^2}{4a_i^2}\right)^{-1} u\right) - \frac{1}{2} u,$$

$$\rho(u) = \prod_{i=1}^{d} \left(1 + \frac{u^2}{4a_i^2}\right)^{1/4} \exp\left(\frac{1}{32} \frac{(u\gamma_i/a_i^2)^2}{1 + \frac{u^2}{4a_i^2}}\right).$$

One can rely on numerical integration techniques to compute the above integral to desired accuracy. In our analysis of the approximation error of Algorithm 5.5, we assume the error from the computation of $f_{A,\gamma}(1)$ is negligible.

Before we conclude this subsection, we present another technique for computation of $f_{A,\gamma}(1)$, which is typically faster than the above approach. This approach was proposed by Kume and Wood [68] and relies on the saddle point density approximation technique. While this approach is faster, the downside of it is that it only provides an approximate estimate of $f_{A,\gamma}(1)$. To explain this method, we first present some facts about non-central chi-squared random variables. The cumulant generating function of a non-central chi-squared random variable with density $f_{A,\gamma}$ is given by

$$K(t) = \sum_{i=1}^{d} \left(-\frac{1}{2} \log\left(1 - \frac{t}{a_i}\right) + \frac{1}{4} \frac{\gamma_i^2}{a_i - t} - \frac{\gamma_i^2}{4a_i} \right) \quad (t < \min_i a_i).$$

The first derivative of $K(t)$ is given by

$$K^{(1)}(t) = \sum_{i=1}^{d} \left(\frac{1}{2} \frac{1}{a_i - t} + \frac{1}{4} \frac{\gamma_i^2}{(a_i - t)^2} \right),$$

and higher derivatives are given by

$$K^{(j)}(t) = \sum_{i=1}^{d} \left(\frac{(j-1)!}{2} \frac{1}{(a_i - t)^j} + \frac{j!}{4} \frac{\gamma_i^2}{(a_i - t)^{j+1}} \right), \quad (j \geq 2).$$

Let $\hat{t}$ be the unique solution in $(-\infty, \min_i a_i)$ to the saddle point equation $K^{(1)}(\hat{t}) = 1$. Kume and Wood [68] show that $\hat{t}$ has finite upper and lower bounds

$$\min_i a_i - \frac{d}{4} - \frac{1}{2} \left(\frac{d^2}{4} + d \max_i \gamma_i^2 \right)^{1/2} \leq \hat{t} \leq \min_i a_i - \frac{1}{4} - \frac{1}{2} \left(\frac{1}{4} + \gamma_{\min}^2 \right)^{1/2},$$

where $\gamma_{\min}$ is equal to γ_{i^*} for $i^* = \operatorname{argmin}_i a_i$. So, to find $\hat{t}$, one can perform grid search in the above range. Given $\hat{t}$, the first-order saddle point density approximation of $f_{A,\gamma}(1)$ is given by

$$\hat{f}_{A,\gamma,1}(1) = \left(2\pi K^{(2)}(\hat{t}) \right)^{-1/2} \exp(K(\hat{t}) - \hat{t}).$$

The second-order saddle point density approximation of $Z_{g,h}(1)$ is given by

$$\hat{f}_{A,\gamma,2}(1) = \hat{f}_{A,\gamma,1}(1)(1 + T),$$

where $T = \frac{1}{8}\hat{\rho}_4 - \frac{5}{24}\hat{\rho}_3^2$, where $\hat{\rho}_j = K^{(j)}(\hat{t}) / (K^{(2)}(\hat{t}))^{j/2}$.

5.8.9.3 Proof of Theorem 5.10

Before we present the proof of the Theorem we present useful intermediate results which we require in our proof.

Intermediate Results

Lemma 5.5 (Lipschitz Continuity)**.** *Consider the problem of linear regression described in Section 5.2.2. Let $\Theta = \{\theta : \theta \in \mathbb{R}^d, \|\theta\|_2 \leq B\}$ and let $\hat{\theta}$ be any estimator which maps the data $D_n = \{(X_i, Y_i)\}_{i=1}^n$ to an element in Θ. Then the risk $R(\hat{\theta}, \theta) = \mathbb{E}_{D_n}\left[\|\hat{\theta}(D_n) - \theta\|_2^2\right]$*

is Lipschitz continuous in its second argument w.r.t ℓ_2 norm over the domain Θ, with Lipschitz constant $4(B+B^2\sqrt{nd})$. Moreover, $R(\hat{\theta}, b\mathbf{e}_1) = \mathbb{E}_{D_n}\left[\|\hat{\theta}(D_n) - b\mathbf{e}_1\|_2^2\right]$ is Lipschitz continuous in b over the domain $[0, B]$, with Lipschitz constant $4(B + B^2\sqrt{n})$.

Proof. Let $R_{\hat{\theta}}(\theta) = R(\hat{\theta}, \theta)$. The gradient of $R_{\hat{\theta}}(\theta)$ with respect to θ is given by

$$\nabla_\theta R_{\hat{\theta}}(\theta) = \mathbb{E}_{D_n}\left[2(\theta - \hat{\theta}(D_n))\right] + \mathbb{E}_{D_n}\left[\|\hat{\theta}(D_n) - \theta\|_2^2 \mathbf{X}^T(\mathbf{Y} - \mathbf{X}\theta)\right],$$

where $\mathbf{X} = [X_1, X_2, \dots X_n]^T, \mathbf{Y} = [Y_1, \dots Y_n]$. The norm of $\nabla_\theta R_{\hat{\theta}}(\theta)$ can be upper bounded as follows

$$\begin{aligned}
\|\nabla_\theta R_{\hat{\theta}}(\theta)\|_2 &\leq \left|\left|\mathbb{E}_{D_n}\left[2(\theta - \hat{\theta}(D_n))\right]\right|\right|_2 + \left|\left|\mathbb{E}_{D_n}\left[\|\hat{\theta}(D_n) - \theta\|_2^2 \mathbf{X}^T(\mathbf{Y} - \mathbf{X}\theta)\right]\right|\right|_2 \\
&\overset{(a)}{\leq} 4B + \mathbb{E}_{D_n}\left[\|\mathbf{X}^T(\mathbf{Y} - \mathbf{X}\theta)\|_2 \|\hat{\theta}(D_n) - \theta\|_2^2\right] \\
&\overset{(b)}{\leq} 4B + 4B^2\mathbb{E}_{D_n}\left[\|\mathbf{X}^T(\mathbf{Y} - \mathbf{X}\theta)\|_2\right] \\
&\leq 4B + 4B^2\sqrt{nd},
\end{aligned}$$

where the first term in (a) follows from the fact that $\theta, \hat{\theta}(X) \in \Theta$ and the second term follows from Jensen's inequality. This shows that $R_{\hat{\theta}}(\theta)$ is Lipschitz continuous over Θ. This finishes the first part of the proof. To show that $R(\hat{\theta}, b\mathbf{e}_1)$ is Lipschitz continuous in b, we use similar arguments. Let $R_{\hat{\theta}}(b) = R(\hat{\theta}, b\mathbf{e}_1)$. Then

$$\begin{aligned}
\left|R'_{\hat{\theta}}(b)\right| &= \left|\left\langle \mathbf{e}_1, \nabla_\theta R_{\hat{\theta}}(\theta)\Big|_{\theta = b\mathbf{e}_1}\right\rangle\right| \\
&\overset{(a)}{\leq} \left|\mathbb{E}_{D_n}\left[2(b - [\hat{\theta}(D_n)]_1)\right]\right| + \left|\left|\mathbb{E}_{D_n}\left[\mathbf{e}_1^T\mathbf{X}^T(\mathbf{Y} - \mathbf{X}\theta)\|\hat{\theta}(D_n) - b\mathbf{e}_1\|_2^2\right]\right|\right|_2 \\
&\leq 4B + 4B^2\mathbb{E}_{D_n}\left[|\mathbf{e}_1^T\mathbf{X}^T(\mathbf{Y} - \mathbf{X}\theta)|\right] \\
&\leq 4B + 4B^2\sqrt{n},
\end{aligned}$$

where (a) follows from our bound for $\|\nabla_\theta R_{\hat{\theta}}(\theta)\|_2$ obtained above. □

Lemma 5.6 (Approximation of risk)**.** *Consider the setting of Lemma 5.5. Let $\hat{\theta}$ be any estimator which maps D_n to an element in Θ. Let $\{D_{n,k}\}_{k=1}^N$ be N independent datasets generated from the linear regression model with true parameter θ. Then with probability at least $1 - \delta$*

$$\left|\frac{1}{N}\sum_{i=1}^N \|\hat{\theta}(D_{n,i}) - \theta\|_2^2 - R_{\hat{\theta}}(\theta)\right| \leq 4B^2\sqrt{\frac{\log\frac{1}{\delta}}{N}}$$

Proof. The proof of the Lemma relies on concentration properties of sub-Gaussian random variables. Let $Z(D_n) = \|\hat{\theta}(D_n) - \theta\|^2$. Note that $R_{\hat{\theta}}(\theta) = \mathbb{E}_{D_n}[Z(D_n)]$. Since $Z(D_n)$ is bounded by $4B^2$, it is a sub-Gaussian random variable. Using Hoeffding bound we get

$$\left|\frac{1}{N}\sum_{i=1}^N Z(D_{n,i}) - \mathbb{E}[Z(D_n)]\right| \leq 4B^2\sqrt{\frac{\log\frac{1}{\delta}}{N}}, \quad \text{w.p} \geq 1 - \delta.$$

□

Main Argument The proof uses exactly the same arguments as in the proof of Theorem 5.9. The only difference between the two proofs are the Lipschitz constants derived in Lemmas 5.3, 5.5. The Lipschitz constant in the case of regression is $O(B+B^2\sqrt{n})$, whereas in the case of finite Gaussian sequence model it is $O(B+B^2)$.

Approximation Error of Algorithm 5.4. There are two causes for error in the optimization oracle described in Algorithm 5.4: (a) grid search and (b) approximate computation of risk $R(\hat{\theta}, b\mathbf{e}_1)$. We now bound the error due to both (a) and (b). From Lemma 5.6 we know that for any estimator $\hat{\theta}_i$ and grid point b_j, the following holds with probability at least $1-\delta$

$$\Big|\frac{1}{N_1}\sum_{k=1}^{N_1}\|\hat{\theta}_i(D_{n,k})-b_j\mathbf{e}_1\|_2^2 - R(\hat{\theta}_i, b_j\mathbf{e}_1)\Big| \leq 4B^2\sqrt{\frac{\log\frac{1}{\delta}}{N_1}}.$$

Taking a union bound over all estimators $\{\hat{\theta}_i\}_{i=1}^T$ and grid points $\{b_j\}_{j=1}^{B/w}$, we can show that with probability at least $1-\delta$, the following holds for all $i\in[T], j\in[B/w]$

$$\Big|\frac{1}{N_1}\sum_{k=1}^{N_1}\|\hat{\theta}_i(D_{n,k})-b_j\mathbf{e}_1\|_2^2 - R(\hat{\theta}_i, b_j\mathbf{e}_1)\Big| \leq 4B^2\sqrt{\frac{\log\frac{BT}{w\delta}}{N_1}}. \tag{5.29}$$

Let $f_{t,\sigma}(b)$ be the actual objective we would like to optimize in iteration t of Algorithm 5.1, which is given by

$$f_{t,\sigma}(b) = \sum_{i=1}^{t-1} R(\hat{\theta}_i, b\mathbf{e}_1) + \sigma b.$$

Let $\hat{f}_{t,\sigma}(b)$ be the approximate objective we are optimizing by replacing $R(\hat{\theta}_i, b\mathbf{e}_1)$ with its approximate estimate. Let b_t^* be a maximizer of $f_{t,\sigma}(b)$ and $b_{t,\text{approx}}^*$ be the maximizer of $\hat{f}_{t,\sigma}(b)$ (which is also the output of Algorithm 5.4). Finally, let $b_{t,\text{NN}}^*$ be the point on the grid which is closest to b_t^*. Using Lemma 5.5 we first show that $f_{t,\sigma}(b)$ is Lipschitz continuous in b. The derivative of $f_{t,\sigma}(b)$ with respect to b is given by

$$f'_{t,\sigma}(b) = \sum_{i=1}^{t-1}\left\langle \mathbf{e}_1, \nabla_\theta R(\hat{\theta}_i,\theta)\Big|_{\theta=b\mathbf{e}_1}\right\rangle + \sigma$$

Using Lemma 5.5, the magnitude of $f'_{t,\sigma}(b)$ can be upper bounded as

$$|f'_{t,\sigma}(b)| \leq 4(t-1)(B+B^2\sqrt{n})+\sigma.$$

This shows that $f_{t,\sigma}(b)$ is Lipschitz continuous in b. We now bound $f_{t,\sigma}(b_t^*) - f_{t,\sigma}(b_{t,\text{approx}}^*)$, the approximation error of the optimization oracle

$$\begin{aligned}
f_{t,\sigma}(b_t^*) &\overset{(a)}{\leq} f_{t,\sigma}(b_{t,\text{NN}}^*) + \left(4t(B + B^2\sqrt{n}) + \sigma\right) w \\
&\overset{(b)}{\leq} \hat{f}_{t,\sigma}(b_{t,\text{NN}}^*) + 4tB^2\sqrt{\frac{\log \frac{BT}{w\delta}}{N_1}} + \left(4t(B + B^2\sqrt{n}) + \sigma\right) w \\
&\overset{(c)}{\leq} \hat{f}_{t,\sigma}(b_{t,\text{approx}}^*) + 4tB^2\sqrt{\frac{\log \frac{BT}{w\delta}}{N_1}} + \left(4t(B + B^2\sqrt{n}) + \sigma\right) w \\
&\overset{(d)}{\leq} f_{t,\sigma}(b_{t,\text{approx}}^*) + 8tB^2\sqrt{\frac{\log \frac{BT}{w\delta}}{N_1}} + \left(4t(B + B^2\sqrt{n}) + \sigma\right) w,
\end{aligned}$$

where (a) follows from Lipschitz property of the loss function and $(b), (d)$ follow from Equation (5.29) and hold with probability at least $1-\delta$ and (c) follows from the optimality of $b_{t,\text{approx}}^*$. This shows that Algorithm 5.4 is a $\left(O\left(TB^2\sqrt{\frac{\log \frac{BT}{w\delta}}{N_1}} + TB(1 + B\sqrt{n})w\right), w\right)$-approximate maximization oracle; that is

$$\alpha = O\left(TB^2\sqrt{\frac{\log \frac{BT}{w\delta}}{N_1}} + TB(1 + B\sqrt{n})w\right), \quad \beta = w.$$

Approximation Error of Algorithm 5.5. There are two sources of approximation error in Algorithm 5.5: (a) computation of mean and normalization constant of FB distribution, and (b) approximation of P_t with its samples. In this analysis we assume that mean and normalization constant of FB distribution can be computed to very high accuracy. So the main focus here is on bounding the error from approximation of P_t.

First, note that since we are using grid search to optimize the maximization problem, the true distribution P_t for which we are supposed to compute the Bayes estimator is a discrete distribution supported on grid points $\{b_1, \dots b_{B/w}\}$. Algorithm 5.5 does not compute the Bayes estimator for P_t. Instead, we generate samples from P_t and use them as a proxy for P_t. Let $\hat{P}_t$ be the empirical distribution obtained by sampling N_2 points from P_t. Let $p_{t,j}$ be the probability mass on grid point b_j. Using Bernstein inequality we can show that the following holds with probability at least $1 - \delta$

$$\forall j \in [B/w] \quad |\hat{p}_{t,j} - p_{t,j}| \leq \sqrt{p_{t,j}\frac{\log \frac{B}{w\delta}}{N_2}}. \tag{5.30}$$

Define estimators $\hat{\theta}'_t, \hat{\theta}_t$ as

$$\hat{\theta}'_t \leftarrow \operatorname*{argmin}_{\hat{\theta} \in \mathcal{D}_G} \mathbb{E}_{b\sim P_t}\left[R(\hat{\theta}, b\mathbf{e}_1)\right], \quad \hat{\theta}_t \leftarrow \operatorname*{argmin}_{\hat{\theta} \in \mathcal{D}_G} \mathbb{E}_{b\sim \hat{P}_t}\left[R(\hat{\theta}, b\mathbf{e}_1)\right].$$

$\hat{\theta}'_t$ is what we ideally want to compute. $\hat{\theta}_t$ is what we end up computing using Algorithm 5.5. We now show that $\hat{\theta}_t$ is an approximate minimizer of the left hand side optimization problem above. To this end, we try to bound the following quantity

$$\mathbb{E}_{b\sim P_t}\left[R(\hat{\theta}_t, b\mathbf{e}_1) - R(\hat{\theta}'_t, b\mathbf{e}_1)\right].$$

Let $f_t(\hat{\theta}) = \mathbb{E}_{b\sim P_t}\left[R(\hat{\theta}, b\mathbf{e}_1)\right]$ and $\hat{f}_t(\hat{\theta}) = \mathbb{E}_{b\sim \hat{P}_t}\left[R(\hat{\theta}, b\mathbf{e}_1)\right]$. We would like to bound the quantity $f_t(\hat{\theta}_t) - f_t(\hat{\theta}'_t)$. Consider the following

$$\begin{aligned} f_t(\hat{\theta}_t) &\overset{(a)}{\leq} \hat{f}_t(\hat{\theta}_t) + \frac{4B^3}{w}\sqrt{\frac{\log\frac{B}{w\delta}}{N_2}} \\ &\overset{(b)}{\leq} \hat{f}_t(\hat{\theta}'_t) + \frac{4B^3}{w}\sqrt{\frac{\log\frac{B}{w\delta}}{N_2}} \\ &\overset{(c)}{\leq} f_t(\hat{\theta}'_t) + \frac{8B^3}{w}\sqrt{\frac{\log\frac{B}{w\delta}}{N_2}}, \end{aligned}$$

where (a) follows from Equation (5.30) and the fact that the risk $R(\hat{\theta}, \theta)$ of any estimator is bounded by $4B^2$, (b) follows since $\hat{\theta}_t$ is a minimizer of $\hat{f}_t$ and (c) follows from Equation (5.30). This shows that with probability at least $1-\delta$, Algorithm 5.5 is an $O\left(\frac{B^3}{w}\sqrt{\frac{\log\frac{B}{w\delta}}{N_2}}\right)$-approximate optimization oracle; that is,

$$\alpha' = O\left(\frac{B^3}{w}\sqrt{\frac{\log\frac{B}{w\delta}}{N_2}}\right).$$

The rest of the proof is same as the proof of Theorem 5.9 and involves substituting the approximation errors computed above in Corollary 5.2.

Minimax Estimator. We now show that $\hat{\theta}_{AVG}$ is an approximate minimax estimator. Instantiating Corollary 5.2 for the reduced statistical game gives us the following bound, which holds with probability at least $1-\delta$

$$\sup_{b\in[0,B]} R(\hat{\theta}_{AVG}, b\mathbf{e}_1) \leq \inf_{\hat{\theta}\in\mathcal{D}_G} \sup_{b\in[0,B]} R(\hat{\theta}, b\mathbf{e}_1) + \tilde{O}\left(\frac{B^2(B\sqrt{n}+1)}{\sqrt{T}} + \alpha + \alpha' + \beta B(B\sqrt{n}+1)\sqrt{T}\right),$$

where we used the fact that the risk $R(\hat{\theta}, b\mathbf{e}_1)$ is $4B(B\sqrt{n}+1)$-Lipschitz continuous w.r.t b. The $\tilde{O}$ notation in the above inequality hides logarithmic factors. Plugging in the values of α, α', β in the above equation gives us

$$\sup_{b\in[0,B]} R(\hat{\theta}_{AVG}, b\mathbf{e}_1) \leq \inf_{\hat{\theta}\in\mathcal{D}_G} \sup_{b\in[0,B]} R(\hat{\theta}, b\mathbf{e}_1) + \tilde{O}\left(\frac{B^2(B\sqrt{n}+1)}{\sqrt{T}}\right).$$

We now convert this bound to a bound on the original statistical game. From Theorem 5.3 we know that $\inf_{\hat{\theta}\in\mathcal{D}_G} \sup_{b\in[0,B]} R(\hat{\theta}, b\mathbf{e}_1) = \inf_{\hat{\theta}\in\mathcal{D}} \sup_{\theta\in\Theta} R(\hat{\theta}, \theta) = R^*$. Since the estimator $\hat{\theta}_{AVG}$ is invariant to transformations of orthogonal group, we have $R(\hat{\theta}_{AVG}, \theta) = R(\hat{\theta}_{AVG}, \|\theta\|_2\mathbf{e}_1)$ for any $\theta \in \Theta$. Using these two results in the above inequality, we get

$$\sup_{\theta\in\Theta} R(\hat{\theta}_{AVG}, \theta) = \sup_{b\in[0,B]} R(\hat{\theta}_{AVG}, b\mathbf{e}_1) \leq R^* + \tilde{O}\left(\frac{B^2(B\sqrt{n}+1)}{\sqrt{T}}\right).$$

This shows that the worst-case risk of $\hat{\theta}_{AVG}$ is close to the minimax risk R^*. This finishes the first part of the proof.

LFP. To prove the second part, we rely on Corollary 5.2. Instantiating it for the reduced statistical game gives us

$$\inf_{\hat{\theta}\in\mathcal{D}_G} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{b\sim P_t}\left[R(\hat{\theta}, b\mathbf{e}_1)\right] \geq R^* - \tilde{O}\left(\frac{B^2(B\sqrt{n}+1)}{\sqrt{T}} + \alpha + \alpha' + \beta B(B\sqrt{n}+1)\sqrt{T}\right).$$

Plugging in the values of α, α', β in the above equation gives us

$$\inf_{\hat{\theta}\in\mathcal{D}_G} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{b\sim P_t}\left[R(\hat{\theta}, b\mathbf{e}_1)\right] \geq R^* - \tilde{O}\left(\frac{B^2(B\sqrt{n}+1)}{\sqrt{T}}\right).$$

From Equation (5.26) we know that P_t is close to $\hat{P}_t$ with high probability. Using this, we can replace P_t in the above bound with $\hat{P}_t$ and obtain the following bound, which holds with probability at least $1-\delta$

$$\inf_{\hat{\theta}\in\mathcal{D}_G} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{b\sim \hat{P}_t}\left[R(\hat{\theta}, b\mathbf{e}_1)\right] \geq R^* - \tilde{O}\left(\frac{B^2(B\sqrt{n}+1)}{\sqrt{T}}\right). \tag{5.31}$$

In the rest of the proof, we show that $\inf_{\hat{\theta}\in\mathcal{D}_G} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{b\sim \hat{P}_t}\left[R(\hat{\theta}, b\mathbf{e}_1)\right] = \inf_{\hat{\theta}} R(\hat{\theta}, \hat{P}_{\text{LFP}})$. From the definition of $\hat{P}_{\text{LFP}}$, we can equivalently rewrite $R(\hat{\theta}, \hat{P}_{\text{LFP}})$ for any estimator $\hat{\theta}$ as

$$R(\hat{\theta}, \hat{P}_{\text{LFP}}) = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{b\sim \hat{P}_t}\left[\mathbb{E}_{\theta\sim U}\left[R(\hat{\theta}, b\theta)\right]\right],$$

where U is the uniform distribution on the $(d-1)$ dimensional unit sphere centered at origin, in $\mathbb{R}^d$. Next, from Lemma 5.2, we know that the Bayes estimator corresponding to any invariant prior is an invariant estimator. Since $\hat{P}_{\text{LFP}}$ is an invariant distribution, we have

$$\inf_{\hat{\theta}\in\mathcal{D}} R(\hat{\theta}, \hat{P}_{\text{LFP}}) = \inf_{\hat{\theta}\in\mathcal{D}_G} R(\hat{\theta}, \hat{P}_{\text{LFP}}) = \inf_{\hat{\theta}\in\mathcal{D}_G} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{b\sim \hat{P}_t}\left[\mathbb{E}_{\theta\sim U}\left[R(\hat{\theta}, b\theta)\right]\right].$$

From Lemma 5.1 we know that for any invariant estimator $\hat{\theta}$, we have $R(\hat{\theta}, \theta_1) = R(\hat{\theta}, \theta_2)$, whenever $\theta_1 \sim \theta_2$. Using this result in the above equation gives us

$$\inf_{\hat{\theta}\in\mathcal{D}} R(\hat{\theta}, \hat{P}_{\text{LFP}}) = \inf_{\hat{\theta}\in\mathcal{D}_G} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{b\sim \hat{P}_t}\left[R(\hat{\theta}, b\mathbf{e}_1)\right].$$

Combining the above result with Equation (5.31) shows that $\hat{P}_{\text{LFP}}$ is approximately least favorable.

5.8.10 Covariance Estimation

5.8.10.1 Proof of Proposition 5.4

In this proof, we rely on permutation invariant functions and a representer theorem for such functions. A function $f : \mathbb{R}^d \to \mathbb{R}$ is called permutation invariant, if for any permutation π and any $X \in \mathbb{R}^d$

$$f(\pi(X)) = f(X).$$

The following proposition provides a representer theorem for such functions.

Proposition 5.9 (Zaheer et al. [111]). *A function $f(X)$ from $\mathbb{R}^d$ to $\mathbb{R}$ is permutation invariant and continuous iff it can be decomposed in the form $\rho(\sum_{i=1}^{d}\phi(X_i))$, for some suitable transformations $\phi : \mathbb{R} \to \mathbb{R}^{d+1}$ and $\rho : \mathbb{R}^{d+1} \to \mathbb{R}$.*

We now prove Proposition 5.4. First note that from Blackwell's theorem we know that there exists a minimax estimator which is just a function of the sufficient statistic, which in this case is the empirical covariance $S_n = \frac{1}{n}\sum_{i=1}^{n} X_i X_i^T$ [see Theorem 2.1 of 50]. So we restrict ourselves to estimators which are functions of S_n. This, together with Theorem 5.2, shows that there is a minimax estimator which is a function S_n and which is invariant under the action of the orthogonal group $\mathbb{O}(d)$. Let $\hat{\Sigma}$ be such an estimator. Since $\hat{\Sigma}$ is an invariant estimator, it satisfies the following equality for any orthogonal matrix V

$$\hat{\Sigma}(VS_nV^T) = V\hat{\Sigma}(S_n)V^T.$$

Setting $V = U^T$ in the above equation, we get $\hat{\Sigma}(S_n) = U\hat{\Sigma}(\Delta)U^T$. Hence, $\hat{\Sigma}$ is completely determined by it's action on diagonal matrices. So, in the rest of the proof we try to understand $\hat{\Sigma}(\Delta)$. Again relying on invariance of $\hat{\Sigma}$ and setting $V = \Delta' U^T$ for some diagonal matrix Δ' with diagonal elements ± 1, we get

$$\hat{\Sigma}(\Delta'\Delta\Delta') = \Delta' U^T \hat{\Sigma}(S_n) U \Delta' \overset{(a)}{=} \Delta' \hat{\Sigma}(\Delta)\Delta',$$

where (a) follows from the fact that $\hat{\Sigma}(S_n) = U\hat{\Sigma}(\Delta)U^T$. Since $\Delta'\Delta\Delta' = \Delta$, the above equation shows that $\Delta'\hat{\Sigma}(\Delta)\Delta' = \hat{\Sigma}(\Delta)$ for any diagonal matrix Δ' with diagonal elements ± 1. This shows that $\hat{\Sigma}(\Delta)$ is a diagonal matrix. Next, we set $V = P_\pi U^T$, where P_π is the permutation matrix corresponding to some permutation π. This gives us

$$\hat{\Sigma}(P_\pi \Delta P_\pi^T) = P_\pi \hat{\Sigma}(\Delta) P_\pi^T.$$

This shows that for any permutation π, $\hat{\Sigma}(\pi(\Delta)) = \pi(\hat{\Sigma}(\Delta))$, where $\pi(\Delta)$ represents permutation of the diagonal elements of Δ. In the rest of the proof, we use the notation Δ_i to denote the i^{th} diagonal entry of Δ and $\hat{\Sigma}_i(\Delta)$ to denote the i^{th} diagonal entry of $\hat{\Sigma}(\Delta)$. The above property of $\hat{\Sigma}$ shows that $\hat{\Sigma}_i(\Delta)$ doesn't depend on the ordering of the elements in $\{\Delta_j\}_{j\neq i}$. This follows by choosing any permutation π which keeps the i^{th} element fixed. Next, by considering the permutation which only exchanges positions 1 and i, we get

$$\hat{\Sigma}_i(\Delta_1, \dots \Delta_i, \dots \Delta_d) = \hat{\Sigma}_1(\Delta_i, \dots \Delta_1, \dots \Delta_d).$$

Thus $\hat{\Sigma}_i$ can be expressed in terms of $\hat{\Sigma}_1$. Represent $\hat{\Sigma}_1$ by $\hat{\Sigma}_0$. Combining the above two properties, we have

$$\hat{\Sigma}_i(\Delta) = \hat{\Sigma}_0(\Delta_i, \{\Delta_j\}_{j\neq i}),$$

where $\{\Delta_j\}_{j\neq i}$ represents the independence of $\hat{\Sigma}_0$ on the ordering of elements $\{\Delta_j\}_{j\neq i}$. Now, consider the function $\hat{\Sigma}_0(\Delta_1, \{\Delta_j\}_{j=2}^d)$. For any fixed a, and $\Delta_1 = a$, $\hat{\Sigma}_0(a, \{\Delta_j\}_{j=2}^d)$ is a permutation invariant function. Using Proposition 5.9, $\hat{\Sigma}_0(a, \{\Delta_j\}_{j=2}^d)$ can be written as

$$\hat{\Sigma}_0(a, \{\Delta_j\}_{j=2}^d) = f_a\left(\sum_{j=2}^{d} g_a(\Delta_j)\right),$$

for some functions f_a, g_a. We overload the notation and define $f_a(x) = f(a, x)$ and $g_a(x) = g(a, x)$. Using this, we can represent $\hat{\Sigma}_i(\Delta)$ as

$$\hat{\Sigma}_i(\Delta) = f\left(\Delta_i, \sum_{j \neq i} g(\Delta_i, \Delta_j)\right),$$

for some functions f, g. There is a small technicality which we ignored while using Proposition 5.9 on $\hat{\Sigma}_0$. Proposition 5.9 only holds for continuous functions. Since $\hat{\Sigma}_0$ is not guaranteed to be continuous, the proposition can't be used on this function. However, this is not an issue because any measurable function is a limit of continuous functions. Since $\hat{\Sigma}_0$ is a measurable function, it can be approximated arbitrarily close in the form of $f_a\left(\sum_{j=2}^{d} g_a(\Delta_j)\right)$.

To conclude the proof of the proposition, we note that

$$\inf_{\hat{\Sigma} \in \mathcal{M}_{\mathcal{D},G}} \sup_{\lambda \in \Xi_G} R(\hat{\Sigma}, \mathrm{Diag}(\lambda)) = \inf_{\hat{\Sigma} \in \mathcal{M}_{f,g}} \sup_{\lambda \in \Xi_G} R(\hat{\Sigma}, \mathrm{Diag}(\lambda)).$$

This is because the minimax estimator can be approximated arbitrarily well using estimators of the form $\hat{\Sigma}_i(\Delta) = f\left(\Delta_i, \sum_{j \neq i} g(\Delta_i, \Delta_j)\right)$ and the fact that the model class has absolutely continuous distributions.

5.8.11 Entropy Estimation

5.8.11.1 Proof of Proposition 5.5

First note that any estimator of entropy is a function of $\hat{P}_n$, which is a sufficient statistic for the problem. This, together with Theorem 5.2, shows that there is a minimax estimator which is a function of $\hat{P}_n$ and which is invariant under the action of permutation group. Let $\hat{f} : \mathbb{R}^d \to \mathbb{R}$ be such an estimator. Since $\hat{f}$ is invariant, it satisfies the following property for any permutation π

$$\hat{f}(\pi(\hat{P}_n)) = \hat{f}(\hat{P}_n).$$

If $\hat{f}(\hat{P}_n)$ is continuous, then Proposition 5.9 shows that it can written as $g\left(\sum_{j=1}^{d} h(\hat{p}_j)\right)$, for some functions $h : \mathbb{R} \to \mathbb{R}^{d+1}, g : \mathbb{R}^{d+1} \to \mathbb{R}$. Even if it is not continuous, since it is a measurable function, it is a limit of continuous functions. So $\hat{f}$ can be approximated arbitrarily close in the form of $g\left(\sum_{j=1}^{d} h(\hat{p}_j)\right)$. This also implies the statistical game in Equation (5.12) can reduced to the following problem

$$\inf_{\hat{f} \in \mathcal{M}_{\mathcal{D},G}} \sup_{P \in \mathcal{P}_G} R(\hat{f}, P) = \inf_{\hat{f} \in \mathcal{M}_{g,h}} \sup_{P \in \mathcal{P}_G} R(\hat{f}, P).$$

5.8.12 Further Experiments

5.8.12.1 Covariance Estimation

In this section, we compare the performance of various estimators at randomly generated Σ's. We use beta distribution to randomly generate Σ's with varying spectral decays and compute the average risks of all the estimators at these Σ's. Figure 5.2 presents the results from this experiment. It can be seen that our estimator has better average case performance than empirical and James Stein estimators.

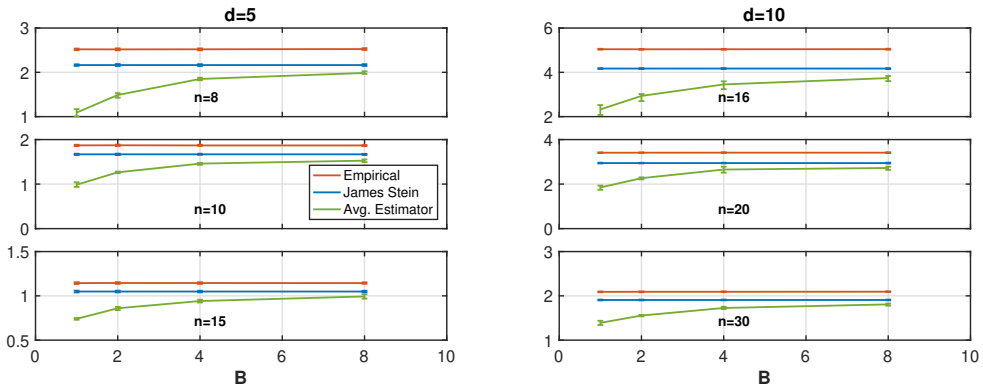


Figure 5.2: Risk of various estimators for covariance estimation evaluated at randomly generated Σ's. We generated multiple Σ's whose eigenvalues are randomly sampled from a Beta distribution with various parameters and averaged the risks of estimators at these Σ's. Plots on the left correspond to $d = 5$ and the plots on the right correspond to $d = 10$.

5.8.12.2 Entropy Estimation

In this section, we compare the performance of various estimators at randomly generated P's. We use beta distribution to randomly generate P's and compute the average risks of all the estimators at these P's. Figure 5.3 presents the results from this experiment.

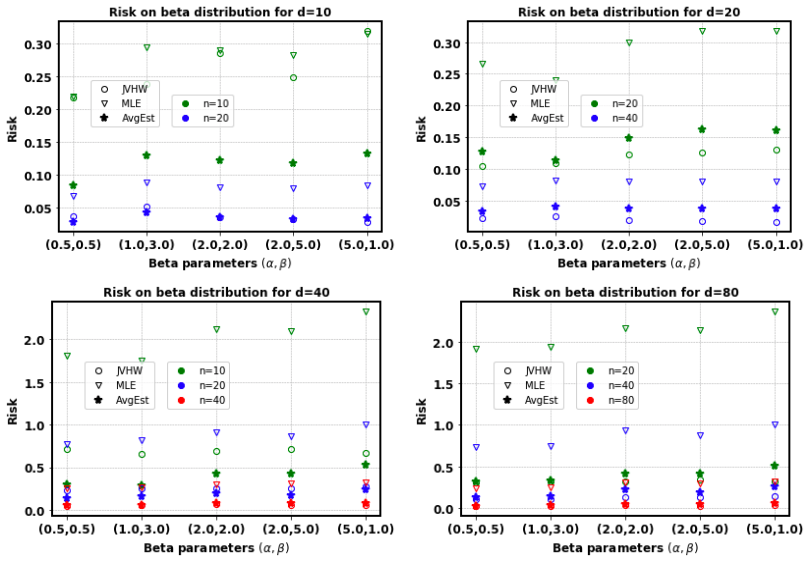


Figure 5.3: Risk of various estimators for entropy estimation evaluated at randomly generated distributions. We generated multiple P's with p_i's sampled from a Beta distribution and averaged the risks of estimators at these P's.

Part IV

Conclusion

6 | Conclusion

In this chapter, we provide an overview of the techniques developed in the thesis, discuss their importance, and present some future research directions. Broadly speaking, we developed three different ideas in the thesis. The first idea builds into a new random walk which has very general applicability and provides strong robustness performance across a wide range of machine learning models in the presence of data poisoning attacks. The second idea provides a new perspective on Euclidean optimization by formulating it over two different manifolds: the Grassmannian and the Multinomial manifold. The third idea solves the classical statistical estimation problem of minimax estimators using latest developments in online learning, providing optimal estimators for fundamental problems for the first time. The first two parts of the thesis are held together by the use of the Grassmannian as the driving force for algorithmic innovation, while the last two parts of the thesis are held together by the use of clever black-box solvers to develop innovative algorithmic techniques with provable guarantees. The latter is a general theme across all the sections of the thesis and highlights the possibilities that open up with the right set of assumptions, here the assumptions being access to right solvers.

6.1 Stochastic optimization for combating data poisoning attacks

In Chapter 2, we present a new algorithm for robust stochastic optimization. We give a very general convergence theorem for this algorithm, identify an important parameter of the analysis (the gap parameter) and experimentally study the robustness properties of our algorithm. We give a modification of our algorithm which can control the robustness of its output by controlling the gap parameter of the loss function it optimizes and discuss the role of k in our algorithm. We also present a very general lemma about the probability of an element picked at random from a Lie group being non-trivially away from it's maximum. We believe that this lemma is very novel, can be adapted to a lot of other settings, and will be useful in future analyses.

While many approaches to stochastic optimization exist in the literature and various defense strategies for data poisoning attacks have been proposed, our approach stands out because of it's general applicability. Apart from the goal of optimization and robustness, our random walk also has the potential to provide privacy properties because of it's extensive use of randomness. Studying the privacy properties of our approach is a promising future direction. We believe that developing algorithms which can address many different

requirements at the same time and work for a vast variety of optimization problems is necessary given the recent explosion in machine learning research. This paper seeks to advance such a research.

6.2 New perspectives on Euclidean optimization

In Chapters 3 and 4, we introduced two new techniques for Euclidean optimization. In the former we formulated Euclidean optimization as a problem on the Grassmannian, and in the latter we formulated it as a problem on the Multinomial manifold. Our approach is novel in two ways. Firstly, it is a novel framework for Euclidean optimization as it introduces a way of using entirely new manifolds for the task. Secondly, it is a novel use of optimization on the used manifolds, since the solution we are seeking does not live on them but the points of the manifolds are just an accessory to finding the solution. The advantage of developing such techniques is that they provide a fresh perspective and an inspiration to develop alternative methods. For example, Algorithms 2.1 and 2.2, developed in Chapter 2, were inspired by these techniques. Hence, while these techniques did not yield immediate practical benefits, they served as an intermediary in the development of the robust stochastic optimization techniques of Chapter 2.

The techniques developed in Chapters 3 and 4 can be further looked upon as an exact method of doing dimension reduction. Dimension reduction is a popular area of research in computer science. Inspired by the results of Johnson and Lindenstrauss [54], many advanced techniques have been built that provide various trade-offs between accuracy and the dimension to which the problem is reduced to [107]. In our techniques, we eliminate the component of loss in accuracy and are able to retrieve the full solution by generating a sequence of smaller-dimensional problems. This provides a new perspective on dimension reduction by exposing some of their geometric underpinnings. The existing approaches to these techniques have mostly been probabilistic in nature, in the sense that the main technical analysis goes via analyzing the randomness used in the dimension reduction process. Our techniques open a new avenue with a geometric perspective by posing the problem on manifolds. In the future, it would be interesting to see what new techniques can be developed by combining the two and find applications to other domains like robustness and privacy.

6.3 Minimax estimators using online learning

In Chapter 5, we introduced an algorithmic approach for constructing minimax estimators, where we attempt to directly solve the min-max statistical game associated with the estimation problem. This is unlike the traditional approach in statistics, where an estimator is first proposed and then its minimax optimality is certified by showing its worst-case risk matches the known lower bounds for the minimax risk. Our algorithm relies on techniques from online non-convex learning for solving the statistical game and requires access to certain optimization subroutines. Given access to these subroutines, our algorithm returns a minimax estimator and a least favorable prior. This reduces the problem of designing minimax estimators to a purely computational question of efficient implementation of these subroutines. While implementing these subroutines is computationally expensive in the

worst case, we showed that one can rely on the structure of the problem to reduce their computational complexity. For the well studied problems of finite Gaussian sequence model and linear regression, we showed that our approach can be used to learn provably minimax estimators in poly(d) time. For problems where provable implementation of the optimization subroutines is computationally expensive, we demonstrated that our framework can still be used together with heuristics to obtain estimators with better performance than existing (up to constant-factor) minimax estimators. We empirically demonstrated this on classical problems such as covariance and entropy estimation. We believe our approach could be especially useful in high-dimensional settings where classical estimators are sub-optimal and not much is known about minimax estimators. In such settings, our approach can provide insights into least favourable priors and aid statisticians in designing minimax estimators.

There are several avenues for future work. The most salient is a more comprehensive understanding of settings where the optimization subroutines can be efficiently implemented. In this work, we have mostly relied on invariance properties of statistical games to implement these subroutines. As described in Section 5.1, there are several other forms of problem structure that can be exploited to implement these subroutines. Exploring these directions can help us construct minimax estimators for several other estimation problems. Another direction for future work would be to modify our algorithm to learn an approximate minimax estimator (*i.e.,* a rate optimal estimator), instead of an exact minimax estimator. There are several reasons why switching to approximate rather than exact minimaxity can be advantageous. First, with respect to our risk tolerance, it may suffice to construct an estimator whose worst-case risk is constant factors worse than the minimax risk. Second, by switching to approximate minimaxity, we believe one can design algorithms requiring significantly weaker optimization subroutines than those required by our current algorithm. Third, the resulting algorithms might be less tailored or over-fit to the specific statistical model assumptions, so that the resulting algorithms will be much more broadly applicable. Towards the last point, we note that our minimax estimators could always be embedded within a model selection sub-routine, so that for any given data-set, one could select from a suite of minimax estimators using standard model selection criteria. Finally, it would be of interest to modify our algorithm to output a single estimator which is simultaneously minimax for various values of n, the number of observations.

Bibliography

[1] https://www.stat.cmu.edu/~ryantibs/convexopt-F13/scribes/lec6.pdf. 3.3

[2] P-A Absil. Riemannian geometry of grassmann manifolds with a view on algorithmic computation. *Acta Applicandae Mathematica*, 80:199–220, 2004. 3.1, 3.1.1, 3.1.2

[3] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, USA, 2007. ISBN 0691132984. 1.1.2, 4.3, 4.1

[4] Naman Agarwal, Alon Gonen, and Elad Hazan. Learning in non-convex games with an optimization oracle. *arXiv preprint arXiv:1810.07362*, 2018. 1.1.4

[5] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009. ISBN 9781139477369. URL https://books.google.com/books?id=nGvI7cOuOOQC. 2, 2.3.1

[6] Arindam Banerjee, Inderjit S Dhillon, Joydeep Ghosh, and Suvrit Sra. Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 6(Sep):1345–1382, 2005. 5.8.7.1

[7] B. Bekka, P. de la Harpe, and A. Valette. *Kazhdan's Property (T)*. New Mathematical Monographs. Cambridge University Press, 2008. ISBN 9781139471084. URL https://books.google.com/books?id=QCftywollBMC. 2.1, 2.4

[8] Thomas Bendokat, Ralf Zimmermann, and P.-A. Absil. A grassmann manifold handbook: basic geometry and computational aspects. *Advances in Computational Mathematics*, 50(1), January 2024. ISSN 1572-9044. doi: 10.1007/s10444-023-10090-8. URL http://dx.doi.org/10.1007/s10444-023-10090-8. 2.2, 3.1

[9] James O Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer Science & Business Media, 1985. 1.1.3, 1.1.3, 5, 5.2

[10] J Calvin Berry. Minimax estimation of a bounded normal mean vector. *Journal of Multivariate Analysis*, 35(1):130–139, 1990. 5

[11] D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999. 1.1.2

[12] PJ Bickel. Minimax estimation of the mean of a normal distribution when the parameter space is restricted. *The Annals of Statistics*, 9(6):1301–1309, 1981. 5, 5.3

[13] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In *Proceedings of the 29th International Coference on International*

Conference on Machine Learning, ICML'12, page 1467–1474, Madison, WI, USA, 2012. Omnipress. ISBN 9781450312851. 1.1.1, 2.6.2

[14] Lucien Birgé. Approximation dans les espaces métriques et théorie de l'estimation. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 65(2):181–237, 1983. 5

[15] Lucien Birgé and Pascal Massart. Rates of convergence for minimum contrast estimators. *Probability Theory and Related Fields*, 97(1-2):113–150, 1993. 5

[16] Lawrence D Brown and R Purves. Measurable selections of extrema. *The annals of statistics*, 1(5):902–912, 1973. 5.8.1, 5.8.1

[17] D. Bump. *Lie Groups*. Graduate Texts in Mathematics. Springer New York, 2013. ISBN 9781461480242. URL `https://books.google.com/books?id=x2W4BAAAQBAJ`. 2.2

[18] Cristina Butucea, Mohamed Ndaoud, Natalia A Stepanova, and Alexandre B Tsybakov. Variable selection with hamming loss. *The Annals of Statistics*, 46(5):1837–1875, 2018. 5

[19] T Tony Cai and Mark G Low. Testing composite hypotheses, hermite polynomials and optimal estimation of a nonsmooth functional. *The Annals of Statistics*, 39(2): 1012–1041, 2011. 5, 5, 5.4

[20] George Casella and William E Strawderman. Estimating a bounded normal mean. *The Annals of Statistics*, pages 870–878, 1981. 5.3

[21] Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006. 1.1.4, 5.1

[22] Moses Charikar, Jacob Steinhardt, and Gregory Valiant. Learning from untrusted data. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, page 47–60, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450345286. doi: 10.1145/3055399.3055491. URL `https://doi.org/10.1145/3055399.3055491`. 2.1

[23] Robert S Chen, Brendan Lucier, Yaron Singer, and Vasilis Syrgkanis. Robust optimization for non-convex objectives. In *Advances in Neural Information Processing Systems*, pages 4705–4714, 2017. 5, 5.2

[24] Siyuan Cheng, Yingqi Liu, Shiqing Ma, and Xiangyu Zhang. Deep feature space trojan attack of neural networks by controlled detoxification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(2):1148–1156, May 2021. doi: 10.1609/aaai.v35i2.16201. URL `https://ojs.aaai.org/index.php/AAAI/article/view/16201`. 1

[25] Antonio Emanuele Cinà, Kathrin Grosse, Ambra Demontis, Sebastiano Vascon, Werner Zellinger, Bernhard A. Moser, Alina Oprea, Battista Biggio, Marcello Pelillo, and Fabio Roli. Wild patterns reloaded: A survey of machine learning security against training data poisoning. *ACM Comput. Surv.*, 55(13s), jul 2023. ISSN 0360-0300. doi: 10.1145/3585385. URL `https://doi.org/10.1145/3585385`. 1, 1.1.1, 2.1

[26] Bertrand S Clarke and Andrew R Barron. Jeffreys' prior is asymptotically least favorable under entropy risk. *Journal of Statistical planning and Inference*, 41(1): 37–60, 1994. 1.1.3

[27] Ambra Demontis, Marco Melis, Maura Pintor, Matthew Jagielski, Battista Biggio, Alina Oprea, Cristina Nita-Rotaru, and Fabio Roli. Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 321–338, Santa Clara, CA, August 2019. USENIX Association. ISBN 978-1-939133-06-9. URL `https://www.usenix.org/conference/usenixsecurity19/presentation/demontis`. 1.1.1, 2.6.2

[28] David L Donoho. Statistical estimation and optimal recovery. *The Annals of Statistics*, pages 238–270, 1994. 5.7.2

[29] David L Donoho, Richard C Liu, and Brenda MacGibbon. Minimax risk over hyperrectangles, and implications. *The Annals of Statistics*, pages 1416–1437, 1990. (document), 5.4, 5.7.1, 5.2

[30] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(61): 2121–2159, 2011. URL `http://jmlr.org/papers/v12/duchi11a.html`. 2.1

[31] Ronen Eldan and Yuanzhi Li. Tinystories: How small can language models be and still speak coherent english?, 2023. 1

[32] Uriel Feige, Yishay Mansour, and Robert Schapire. Learning and inference in the presence of corrupted inputs. In *Conference on Learning Theory*, pages 637–657, 2015. 5

[33] Ji Feng, Qi-Zhi Cai, and Zhi-Hua Zhou. Learning to confuse: Generating training time adversarial data with auto-encoder. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper_files/paper/2019/file/1ce83e5d4135b07c0b82afffbe2b3436-Paper.pdf`. 1

[34] Thomas S Ferguson. *Mathematical statistics: A decision theoretic approach*, volume 1. Academic press, 2014. 5

[35] J. Ferrer, MI. Garćia, and F. Puerta. Differentiable families of subspaces. *Linear Algebra and its Applications*, 199:229–252, 1994. ISSN 0024-3795. doi: https://doi.org/10.1016/0024-3795(94)90351-4. URL `https://www.sciencedirect.com/science/article/pii/0024379594903514`. Special Issue Honoring Ingram Olkin. 3.1

[36] Liam Fowl, Micah Goldblum, Ping-yeh Chiang, Jonas Geiping, Wojciech Czaja, and Tom Goldstein. Adversarial examples make strong poisons. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 30339–30351. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper_files/paper/2021/file/fe87435d12ef7642af67d9bc82a8b3cd-Paper.pdf`. 1

[37] Yoav Freund and Robert E Schapire. Game theory, on-line prediction and boosting. In *COLT*, volume 96, pages 325–332. Citeseer, 1996. 5

[38] J. Gallier and J. Quaintance. *Differential Geometry and Lie Groups: A Computational Perspective*. Geometry and Computing. Springer International Publishing, 2020. ISBN 9783030460402. URL `https://books.google.com/books?id=K3r3DwAAQBAJ`. 2.2

[39] Guillaume Garrigos and Robert M. Gower. Handbook of convergence theorems for (stochastic) gradient methods, 2024. 2

[40] MN Ghosh. Uniform approximation of minimax point estimates. *The Annals of Mathematical Statistics*, pages 1031–1047, 1964. 1.1.3

[41] Robert M. Gower, Othmane Sebbouh, and Nicolas Loizou. SGD for structured non-convex functions: Learning rates, minibatching and interpolation. In Arindam Banerjee and Kenji Fukumizu, editors, *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pages 1315–1323. PMLR, 2021. URL `http://proceedings.mlr.press/v130/gower21a.html`. 2.1

[42] Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. Textbooks are all you need, 2023. 1

[43] Kartik Gupta, Arun Sai Suggala, Adarsh Prasad, Praneeth Netrapalli, and Pradeep Ravikumar. Learning minimax estimators via online learning, 2020. URL `https://arxiv.org/abs/2006.11430`. 4

[44] Venkatesan Guruswami and Prasad Raghavendra. Hardness of learning halfspaces with noise. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 543–552, 2006. doi: 10.1109/FOCS.2006.33. 2

[45] Anders Hald. The size of bayes and minimax tests as function of the sample size and the loss ratio. *Scandinavian Actuarial Journal*, 1971(1-2):53–73, 1971. 1.1.3

[46] JA Hartigan. Asymptotic normality of posterior distributions. In *Bayes theory*, pages 107–118. Springer, 1983. 5.4

[47] T. Hastie, R. Tibshirani, and J.H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics. Springer, 2009. ISBN 9780387848846. URL `https://books.google.com/books?id=eBSgoAEACAAJ`. 2.6.1

[48] Elad Hazan. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016. 1.1.4, 5.1

[49] S. Helgason. *Groups and Geometric Analysis: Integral Geometry, Invariant Differential Operators, and Spherical Functions*. Mathematical Surveys and Mono-

graphs. American Mathematical Society, 2022. ISBN 9780821832110. URL https://books.google.com/books?id=ThZuEAAAQBAJ. 2.5

[50] I. A. Ibragimov and R. Z. Has'minskii. *Statistical Estimation: Asymptotic Theory*. springer, New York, 1981. 5, 5.5, 5.8.10.1

[51] Jean-Pierre Imhof. Computing the distribution of quadratic forms in normal variables. *Biometrika*, 48(3/4):419–426, 1961. 5.4, 5.8.9.2

[52] William James and Charles Stein. Estimation with quadratic loss. In *Breakthroughs in statistics*, pages 443–460. Springer, 1992. 5.7.4

[53] Jiantao Jiao, Kartik Venkat, Yanjun Han, and Tsachy Weissman. Minimax estimation of functionals of discrete distributions. *IEEE Transactions on Information Theory*, 61(5):2835–2885, 2015. 5, 5, 5.7.5, 5.6

[54] William B. Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into hilbert space. *Contemporary mathematics*, 26:189–206, 1984. URL https://api.semanticscholar.org/CorpusID:117819162. 6.2

[55] Iain M Johnstone. Function estimation and gaussian sequence models. *Unpublished manuscript*, 2(5.3), 2002. 1.1.3, 5

[56] Iain M Johnstone. Gaussian estimation: Sequence and wavelet models. *Unpublished manuscript*, 2011. 5.3

[57] Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005. 1.1.4

[58] Kirthevasan Kandasamy, Karun Raju Vysyaraju, Willie Neiswanger, Biswajit Paria, Christopher R. Collins, Jeff Schneider, Barnabas Poczos, and Eric P Xing. Tuning Hyperparameters without Grad Students: Scalable and Robust Bayesian Optimisation with Dragonfly. *arXiv preprint arXiv:1903.06694*, 2019. 5.5

[59] Peter J Kempthorne. Numerical specification of discrete least favorable prior distributions. *SIAM Journal on Scientific and Statistical Computing*, 8(2):171–184, 1987. 1.1.3

[60] Ahmed Khaled and Peter Richtárik. Better theory for SGD in the nonconvex world. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=AU4qHN2VkS. Survey Certification. 2.1

[61] Roni Khardon and Gabriel Wachman. Noise tolerant variants of the perceptron algorithm. *J. Mach. Learn. Res.*, 8:227–248, may 2007. ISSN 1532-4435. 2

[62] Jack Kiefer et al. Invariance, minimax sequential estimation, and continuous time processes. *The Annals of Mathematical Statistics*, 28(3):573–601, 1957. 5.2, 5.2

[63] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL https://api.semanticscholar.org/CorpusID:6628106. 2.1

[64] S. Kobayashi and K. Nomizu. *Foundations of Differential Geometry, Volume 2*. Foundations of Differential Geometry [by] Shoshichi Kobayashi and Katsumi Nomizu. Wi-

ley, 1963. ISBN 9780470496480. URL https://books.google.co.in/books?id=603vAAAAMAAJ. 3.1

[65] S. Kobayashi and K. Nomizu. *Foundations of Differential Geometry, Volume 1*. Wiley Classics Library. Wiley, 1996. ISBN 9780471157335. URL https://books.google.co.in/books?id=ss7bEAAAQBAJ. 3.1

[66] E. Kowalski. *An Introduction to Expander Graphs*. Collection SMF / Cours spécialisés. Société Mathématique de France, 2019. ISBN 9782856298985. URL https://books.google.com/books?id=BkmAxQEACAAJ. 2

[67] Walid Krichene, Maximilian Balandat, Claire Tomlin, and Alexandre Bayen. The hedge algorithm on a continuum. In *International Conference on Machine Learning*, pages 824–832, 2015. 1.1.4, 5.1

[68] Alfred Kume and Andrew TA Wood. Saddlepoint approximations for the bingham and fisher–bingham normalising constants. *Biometrika*, 92(2):465–476, 2005. 5.4, 5.4, 5.8.9.2

[69] V. Lakshmibai. *Flag Varieties: An Interplay of Geometry, Combinatorics, and Representation Theory*. Texts and Readings in Mathematics. Hindustan Book Agency, 2009. ISBN 9789386279415. URL https://books.google.co.in/books?id=yfJdDwAAQBAJ. 2.7.8

[70] Lucien Le Cam. *Asymptotic methods in statistical decision theory*. Springer Science & Business Media, 2012. 5

[71] J. Lee. *Introduction to Topological Manifolds*. Graduate Texts in Mathematics. Springer New York, 2010. ISBN 9781441979407. URL https://books.google.co.in/books?id=ZQVGAAAAQBAJ. 1.1.2

[72] J. Lee. *Introduction to Smooth Manifolds*. Graduate Texts in Mathematics. Springer New York, 2012. ISBN 9781441999825. URL https://books.google.co.in/books?id=xygVcKGPsNwC.

[73] J.M. Lee. *Introduction to Riemannian Manifolds*. Graduate Texts in Mathematics. Springer International Publishing, 2019. ISBN 9783319917542. URL https://books.google.co.in/books?id=UIPltQEACAAJ. 1.1.2, 3.1.1

[74] Erich L Lehmann and George Casella. *Theory of point estimation*. Springer Science & Business Media, 2006. 5

[75] Yang Liu, Jiahuan Cao, Chongyu Liu, Kai Ding, and Lianwen Jin. Datasets for large language models: A comprehensive survey, 2024. 1

[76] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X*, page 182–199, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-58606-5. doi: 10.1007/978-3-030-58607-2_11. URL https://doi.org/10.1007/978-3-030-58607-2_11. 1

[77] Alex Luedtke, Marco Carone, Noah Simon, and Oleg Sofrygin. Learning to learn from data: Using deep adversarial learning to construct optimal statistical procedures. *Science Advances*, 6(9), 2020. doi: 10.1126/sciadv.aaw2140. URL `https://advances.sciencemag.org/content/6/9/eaaw2140`. 1.1.3

[78] Éric Marchand and François Perron. On the minimax estimator of a bounded normal mean. *Statistics & probability letters*, 58(4):327–333, 2002. 5, 5.3

[79] Kanti V Mardia and Peter E Jupp. *Directional statistics*, volume 494. John Wiley & Sons, 2009. 5.8.7.1

[80] G. Margulis. Explicit constructions of concentrators. *Problemy Peredachi Informatsii*, 9(4):71–80, 1973. 2.4.1

[81] H Brendan McMahan. A survey of algorithms and analysis for adaptive online learning. *The Journal of Machine Learning Research*, 18(1):3117–3166, 2017. 1.1.4

[82] Marco Melis, Ambra Demontis, Maura Pintor, Angelo Sotgiu, and Battista Biggio. secml: A python library for secure and explainable machine learning. *arXiv preprint arXiv:1912.10013*, 2019. (document), 2.6.2, 2.2

[83] Stephen D. Miller and Ramarathnam Venkatesan. Spectral analysis of pollard rho collisions. In Florian Hess, Sebastian Pauli, and Michael E. Pohst, editors, *Algorithmic Number Theory, 7th International Symposium, ANTS-VII, Berlin, Germany, July 23-28, 2006, Proceedings*, volume 4076 of *Lecture Notes in Computer Science*, pages 573–581. Springer, 2006. doi: 10.1007/11792086_40. URL `https://doi.org/10.1007/11792086_40`. 2.4.1

[84] Eric Moulines and Francis Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL `https://proceedings.neurips.cc/paper_files/paper/2011/file/40008b9a5380fcacce3976bf7c08af5b-Paper.pdf`. 2.1

[85] L. Nachbin. *The Haar Integral*. University series in higher mathematics. R. E. Krieger Publishing Company, 1976. ISBN 9780882753744. URL `https://books.google.com/books?id=8YspAQAAMAAJ`. 2.2

[86] Wayne Nelson. Minimax solution of statistical decision problems by iteration. *The Annals of Mathematical Statistics*, pages 1643–1657, 1966. 1.1.3

[87] Maria-Irina Nicolae, Mathieu Sinn, Minh Ngoc Tran, Beat Buesser, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, Ian Molloy, and Ben Edwards. Adversarial robustness toolbox v1.2.0. *CoRR*, 1807.01069, 2018. URL `https://arxiv.org/pdf/1807.01069`. 2.6.3, 2.6.3

[88] Yury Polyanskiy and Yihong Wu. Dualizing le cam's method, with applications to estimating the unseens. *arXiv preprint arXiv:1902.05616*, 2019. 5.4

[89] Adarsh Prasad, Arun Sai Suggala, Sivaraman Balakrishnan, and Pradeep Ravikumar.

Robust estimation via robust gradient estimation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 82(3):601–627, 2020. 2.1

[90] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), jan 2020. ISSN 1532-4435. 1

[91] J.D. Rogawski and A. Lubotzky. *Discrete Groups, Expanding Graphs and Invariant Measures*. Progress in Mathematics. Birkhäuser Basel, 1994. ISBN 9783764350758. URL `https://books.google.com/books?id=aNURlzNuotEC`. 2.2, 2.4.1

[92] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. Hidden trigger backdoor attacks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34 (07):11957–11965, Apr. 2020. doi: 10.1609/aaai.v34i07.6871. URL `https://ojs.aaai.org/index.php/AAAI/article/view/6871`. 1

[93] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. Hidden trigger backdoor attacks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11957–11965, 2020. (document), 1.1.1, 2.4, 2.6.3

[94] M.R. Sepanski. *Compact Lie Groups*. Graduate Texts in Mathematics. Springer New York, 2006. ISBN 9780387302638. URL `https://books.google.com/books?id=F3NgD_25OOsC`. 2.2

[95] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL `http://jmlr.org/papers/v15/srivastava14a.html`. 2.1

[96] Arun Sai Suggala and Praneeth Netrapalli. Online non-convex learning: Following the perturbed leader is optimal. In *Algorithmic Learning Theory*, pages 845–861, 2020. 1.1.4, 1.1.4, 1.1.4, 5.1, 5.8.3

[97] Y. Sun, J. Gao, X. Hong, B. Mishra, and B. Yin. Heterogeneous tensor decomposition for clustering via manifold optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(3):476–489, 2016. 4.1

[98] Zhiyi Tian, Lei Cui, Jie Liang, and Shui Yu. A comprehensive survey on poisoning attacks and countermeasures in machine learning. *ACM Comput. Surv.*, 55(8), dec 2022. ISSN 0360-0300. doi: 10.1145/3551636. URL `https://doi.org/10.1145/3551636`. 1.1.1, 2.1

[99] Alexandre B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer Publishing Company, Incorporated, 1st edition, 2008. ISBN 0387790519, 9780387790510. 5

[100] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Clean-label backdoor attacks. 2018. (document), 1.1, 1.2, 1.1.1, 2.6.3, 2.3

[101] A. W. van der Vaart. *Asymptotic Statistics*. Cambridge Series in Statistical

and Probabilistic Mathematics. Cambridge University Press, 1998. doi: 10.1017/CBO9780511802256. 5

[102] G. Valiant and P. Valiant. The power of linear estimators. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 403–412, Oct 2011. doi: 10.1109/FOCS.2011.81. 5

[103] John Von Neumann, Oskar Morgenstern, and Harold William Kuhn. *Theory of games and economic behavior (commemorative edition)*. Princeton university press, 2007. 1.1.3

[104] Abraham Wald. Statistical decision functions. *The Annals of Mathematical Statistics*, pages 165–205, 1949. 1.1.3, 5.8.4.3

[105] Jon A Wellner. Maximum likelihood in modern times: the ugly, the bad, and the good. 2015. URL `https://www.stat.washington.edu/jaw/RESEARCH/TALKS/LeCam-v2.pdf`. 5

[106] Robert A Wijsman. Invariant measures on groups and their use in statistics. *Lecture Notes-Monograph Series*, 14:i–218, 1990. 5.8.5.1, 5.8.5.2

[107] David P. Woodruff. Sketching as a tool for numerical linear algebra. *Found. Trends Theor. Comput. Sci.*, 10(1–2):1–157, oct 2014. ISSN 1551-305X. doi: 10.1561/0400000060. URL `https://doi.org/10.1561/0400000060`. 6.2

[108] Yihong Wu and Pengkun Yang. Minimax rates of entropy estimation on large alphabets via best polynomial approximation. *IEEE Transactions on Information Theory*, 62(6):3702–3720, 2016. 5

[109] Yuhong Yang and Andrew Barron. Information-theoretic determination of minimax rates of convergence. *Annals of Statistics*, pages 1564–1599, 1999. 5

[110] EB Yanovskaya. Infinite zero-sum two-person games. *Journal of Soviet Mathematics*, 2(5):520–541, 1974. 1.1.3

[111] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in neural information processing systems*, pages 3391–3401, 2017. 5.9