Interactions Between Learning and Decision Making

by

Theja Tulabandhula

B.Tech., and M.Tech., IIT Kharagpur (2009)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2014

Signature redacted

Author ....
Department of Electrical Engineering and Computer Science
August 15, 2014

Signature redacted

Certified by ....
Cynthia Rudin
Associate Professor
Thesis Supervisor

Signature redacted

Accepted by ....
Leslie A. Kolodziejski
Chairman, Department Committee on Graduate Theses

Interactions Between Learning and Decision Making

by

Theja Tulabandhula

Submitted to the Department of Electrical Engineering and Computer Science
on August 15, 2014, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

We quantify the effects of learning and decision making on each other in three parts. In the first part, we look at how knowledge about decision making can influence learning. Let the decision cost be the amount spent by the practitioner in executing a policy. If we have prior knowledge about this cost, for instance that it should be low, then this knowledge can help restrict the hypothesis space for learning, which can help with its generalization. We derive a suite of theoretical generalization bounds and an algorithm for this setting.

In the second part, we look at how knowledge about learning can influence decision making. We study this in the context of robust optimization. Taking the uncertainty of learning the right model into account, we derive multiple probabilistic guarantees on the robustness of the resulting policy.

In the last part, we explore the interactions between learning and decision making in depth for two applications. The first application is in the area of power grid maintenance and the second is in the area of professional racing. We provide tailored solutions for modeling, predicting and making decisions in each context.

Thesis Supervisor: Cynthia Rudin
Title: Associate Professor

Acknowledgments

Funding for this project comes in part from a Fulbright Science and Technology Fellowship, a Xerox-MIT Fellowship, a Ford-MIT Alliance grant, an award from the Solomon Buchsbaum Research Fund, and NSF grant IIS-1053407.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The topic of this thesis is the study of the effects of machine learning on decision making and vice versa. In a traditional workflow one performs machine learning using historical data and then uses the constructed prediction model to forecast for a current set of features. These forecasted values are fed into a one shot decision making problem to come up with a policy. The policy performs well if it is comparable to the policy which can be obtained in hindsight (i.e., when the true values realize). We move away from this traditional workflow and propose several interesting ways to look at learning and decision making at the same time.

The first three chapters investigate learning in the setting where we have some sort of prior knowledge. In particular, we can have knowledge about some aspects of a decision making problem which is parameterized by the outputs of the learning step. In this setting, we want to say something about generalization of learning if we know that the decision optimal value lies in a known interval. We also consider a related case where such knowledge can come from unlabeled examples directly. The fourth chapter investigates decision making in the setting where we know something about the learning that parameterizes the decision making formulation. In particular, we are able to come up with guarantees on decision feasibility to unknown future labels when we know that there is going to be estimation and approximation error in the learning step. In the following paragraphs, we highlight each of the chapters in some more detail.

In Chapter 2, we propose a way to align statistical modeling with decision making, which we call the "Machine Learning with Operational Costs (MLOC)" framework. We provide a method that propagates the uncertainty in predictive modeling to the uncertainty in operational cost, where operational cost is the amount spent by the practitioner in solving the problem. The method allows us to explore the range of operational costs associated with the set of reasonable statistical models, so as to provide a useful way for practitioners to understand uncertainty. To do this, the operational cost is cast as a regularization term in a learning algorithm's objective function, allowing either an optimistic or pessimistic view of possible costs, depending on the regularization parameter. From another perspective, if we have prior knowledge about the operational cost, for instance that it should be low, this knowledge can help to restrict the hypothesis space, and can help with generalization. We provide a theoretical generalization bound for this scenario. We also show that learning with operational costs is related to robust optimization.

In Chapter 3, we present a new application and covering number bound for the framework presented in Chapter 2, which is an exploratory form of decision theory. In this work, we use the MLOC framework to study a problem that has implications for power grid reliability and maintenance, called the *Machine Learning and Traveling Repairman Problem* (ML&TRP). The goal of the ML&TRP is to determine a route for a "repair crew," which repairs nodes on a graph. The repair crew aims to minimize the cost of failures at the nodes, but as in many real situations, the failure probabilities are not known and must be estimated. The MLOC framework allows us to understand how this uncertainty influences the repair route. We also present new covering number generalization bounds for the MLOC framework.

In Chapter 4, we consider a supervised learning setting where side knowledge is provided about the labels of unlabeled examples. One of the ways such a side knowledge can arise is through knowledge about an associated decision making problem. This was dealt with in Chapters 2 and 3. So, here we consider other sources of side knowledge. This side knowledge has the effect of reducing the hypothesis space, leading to tighter generalization bounds, and thus possibly better generalization. We consider several types of side knowledge, the first leading to linear and polygonal constraints on the hypothesis space, the second leading to quadratic constraints, and the last leading to conic constraints. We show how different types of domain knowledge can lead directly to these kinds of side knowledge. We prove bounds on complexity measures of the hypothesis space for quadratic and conic side knowledge, and show that these bounds are tight in a specific sense for the quadratic case.

Our goal in Chapter 5 is to build robust optimization problems for making decisions based on complex data from the past. In robust optimization (RO) generally, the goal is to create a policy for decision-making that is robust to our uncertainty about the future. In particular, we want our policy to best handle the the worst possible situation that could arise, out of an *uncertainty set* of possible situations. Classically, the uncertainty set is simply chosen by the user, or it might be estimated in overly simplistic ways with strong assumptions; whereas in this work, we learn the uncertainty set from data collected in the past. The past data are drawn randomly from an (unknown) possibly complicated high-dimensional distribution. We propose a new uncertainty set design and show how tools from statistical learning theory can be employed to provide probabilistic guarantees on the robustness of the policy.

Our goal in Chapter 6 is to design a prediction and decision system for real-time use during a professional car race. In designing a knowledge discovery process for racing, we faced several challenges that were overcome only when domain knowledge of racing was carefully infused within statistical modeling techniques. In this paper, we describe how we leveraged expert knowledge of the domain to produce a real-time decision system for tire changes within a race. Our forecasts have the potential to impact how racing teams can optimize strategy, by making tire-change decisions to benefit their rank position. Our work significantly expands previous research on sports analytics, as it is the only work on analytical methods for within-race prediction and decision making for professional car racing.

Each of the above chapters is written such that it introduces the problem and its background, formalizes it, proposes algorithms if applicable and shows the related proofs and experiments.

Chapter 2

Machine Learning with Operational Costs

2.1 Introduction

Machine learning algorithms are used to produce predictions, and these predictions are often used to make a policy or plan of action afterwards, where there is a cost to implement the policy. In this work, we would like to understand how the uncertainty in predictive modeling can translate into the uncertainty in the cost for implementing the policy. This would help us answer questions like:

Q1. "What is a reasonable amount to allocate for this task so we can react best to whatever nature brings?"

Q2. "Can we produce a reasonable probabilistic model, supported by data, where we might expect to pay a specific amount?"

Q3. "Can our intuition about how much it will cost to solve a problem help us produce a better probabilistic model?"

The three questions above cannot be answered by standard decision theory, where the goal is to produce a single policy that minimizes expected cost. These questions also cannot be answered by robust optimization, where the goal is to produce a single policy that is robust to the uncertainty in nature. Those paradigms produce a single policy decision that takes uncertainty into account, and the chosen policy might not be a best response policy to any realistic situation. In contrast, our goal is to understand the uncertainty and how to react to it, using policies that would be best responses to individual situations.

There are many applications in which this method can be used. For example, in scheduling staff for a medical clinic, predictions based on a statistical model of the number of patients might be used to understand the possible policies and costs for staffing. In traffic flow problems, predictions based on a model of the forecasted traffic might be useful for determining load balancing policies on the network and their associated costs. In online advertising, predictions based on models for the payoff and ad-click rate might be used to understand policies for when the ad should be displayed and the associated revenue.

In order to propagate the uncertainty in modeling to the uncertainty in costs, we introduce what we call the *simultaneous process*, where we explore the range of predictive models and corresponding policy decisions at the same time. The simultaneous process was named to contrast with a more traditional *sequential process*, where first, data are input into a statistical algorithm to produce a predictive model, which makes recommendations for the future, and second, the user develops a plan of action and projected cost for implementing the policy. The sequential process is commonly used in practice, even though there may actually be a whole class of models that could be relevant for the policy decision problem. The sequential process essentially assumes that the probabilistic model is "correct enough" to make a decision that is "close enough."

In the simultaneous process, the machine learning algorithm contains a regularization term encoding the policy and its associated cost, with an adjustable regularization parameter. If there is some uncertainty about how much it will cost to solve the problem, the regularization parameter can be swept through an interval to find a range of possible costs, from optimistic to pessimistic. The method then produces the most likely scenario for each value of the cost. This way, by looking at the full range of the regularization parameter, we sweep out costs for all of the reasonable probabilistic models. This range can be used to determine how much might be reasonably allocated to solve the problem.

Having the full range of costs for reasonable models can directly answer the question in the first paragraph regarding allocation, "What is a reasonable amount to allocate for this task so we can react best to whatever nature brings?" One might choose to allocate the maximum cost for the set of reasonable predictive models for instance. The second question above is "Can we produce a reasonable probabilistic model, supported by data, where we might expect to pay a specific amount?" This is an important question, since business managers often like to know if there is some scenario/decision pair that is supported by the data, but for which the operational cost is low (or high); the simultaneous process would be able to find such scenarios directly. To do this, we would look at the setting of the regularization parameter that resulted in the desired value of the cost, and then look at the solution of the simultaneous formulation, which gives the model and its corresponding policy decision.

Let us consider the third question above, which is "Can our intuition about how much it will cost to solve a problem help us produce a better probabilistic model?" The regularization parameter can be interpreted to regulate the strength of our belief in the operational cost. If we have a strong belief in the cost to solve the problem, and if that belief is correct, this will guide the choice of regularization parameter, and will help with prediction. In many real scenarios, a practitioner or domain expert might truly have a prior belief on the cost to complete a task. Arguably, a manager having this more grounded type of prior belief is much more natural than, for instance, the manager having a prior belief on the ℓ_2 norm of the coefficients of a linear model, or the number of nonzero coefficients in the model. Being able to encode this type of prior belief on cost could potentially be helpful for prediction: as with other types of prior beliefs, it can help to restrict the hypothesis space and can assist with generalization. In this work, we show that the restricted hypothesis spaces resulting from our method can often be bounded by an intersection of an an ℓ_q ball with a halfspace - and this is true for many different types of decision problems. We analyze the complexity of this type of hypothesis space with a technique based on Maurey's Lemma [Barron, 1993, Zhang, 2002] that leads eventually to a counting problem, where we calculate the number of integer points within a polyhedron in order to obtain a covering number bound.

The operational cost regularization term can be the optimal value of a complicated optimization problem, like a scheduling problem. This means we will need to solve an optimization problem each time we evaluate the learning algorithm's objective. However, the practitioner must be able to solve that problem anyway in order to develop a plan of action; it is the same problem they need to solve in the traditional sequential process, or using standard decision theory. Since the decision problem is solved only on data from the present, whose labels are not yet known, solving the decision problem may not be difficult, especially if the number of unlabeled examples is small. In that case, the method can still scale up to huge historical data sets, since the historical data factors into the training error term but not the new regularization term, and both terms can be computed. An example is to compute a schedule for a day, based on factors of the various meetings on the schedule that day. We can use a very large amount of past meeting-length data for the training error term, but then we use only the small set of possible meetings coming up that day to pass into the scheduling problem. In that case, both the training error term and the regularization term are able to be computed, and the objective can be minimized.

The simultaneous process is a type of decision theory. To give some background, there are two types of relevant decision theories: normative (which assumes full information, rationality and infinite computational power) and descriptive (models realistic human behavior). Normative decision theories that address decision making under uncertainty can be classified into those based on ignorance (using no probabilistic information) and those based on risk (using probabilistic information). The former include maximax, maximin (Wald), minimax regret (Savage), criterion of realism (Hurwicz), equally likely (Laplace) approaches. The latter include utility based expected value and bayesian approaches (Savage). Info-gap, Dempster-Shafer, fuzzy logic, and possibility theories offer non-probabilistic alternatives to probability in Bayesian/expected value theories [French, 1986, Hansson, 1994].

The simultaneous process does not fit into any of the decision theories listed above. For instance, a core idea in the Bayesian approach is to choose a single policy that maximizes expected utility, or minimizes expected cost. Our goal is not to find a single policy that is useful on average. In contrast, our goal is to trace out a path of models, their specific (not average) optimal-response policies, and their costs. The policy from the Bayesian approach may not correspond to the best decision for any particular single model, whereas that is something we want in our case. We trace out this path by changing our prior belief on the operational cost (that is, by changing the strength of our regularization term). In Bayesian decision theory, the prior is over possible probabilistic models, rather than on possible costs as in this paper. Constructing this prior over possible probabilistic models can be challenging, and the prior often ends up being chosen arbitrarily, or as a matter of convenience. In contrast, we assume only an unknown probability measure over the data, and the data itself defines the possible probabilistic models for which we compute policies.

Maximax (optimistic) and maximin (pessimistic) decision approaches contrast with the Bayesian framework and do not assume a distribution on the possible probabilistic models. In Section 2.4 we will discuss how these approaches are related to the simultaneous process. They overlap with the simultaneous process but not completely. Robust optimization is a maximin approach to decision making, and the simultaneous process also differs in principle from robust optimization. In robust optimization, one would generally need to allocate much more than is necessary for any single realistic situation, in order to produce a policy that is robust to almost all situations. However, this is not always true; in fact, we show in this work that in some circumstances, while sweeping through the regularization parameter, one of the results produced by the simultaneous process is the same as the one coming from robust optimization.

We introduce the sequential and simultaneous processes in Section 2.2. In Section 2.3, we give several examples of algorithms that incorporate these operational costs.

In doing so, we provide answers for the first two questions Q1 and Q2 above, with respect to specific problems. Our first example application is a staffing problem at a medical clinic, where the decision problem is to staff a set of stations that patients must complete in a certain order. The time required for patients to complete each station is random and estimated from past data. The second example is a real-estate purchasing problem, where the policy decision is to purchase a subset of available properties. The values of the properties need to be estimated from comparable sales. The third example is a call center staffing problem, where we need to create a staffing policy based on historical call arrival and service time information. A fourth example is the "Machine Learning and Traveling Repairman Problem" (ML&TRP) where the policy decision is a route for a repair crew. As mentioned above, there is a large subset of problems that can be formulated using the simultaneous process that have a special property: they are equivalent to robust optimization (RO) problems. Section 2.4 discusses this relationship and provides, under specific conditions, the equivalence of the simultaneous process with RO. Robust optimization, when used for decision-making, does not usually include machine learning, nor any other type of statistical model, so we discuss how a statistical model can be incorporated within an uncertainty set for an RO. Specifically, we discuss how different loss functions from machine learning correspond to different uncertainty sets. We also discuss the overlap between RO and the optimistic and pessimistic versions of the simultaneous process.

We consider the implications of the simultaneous process on statistical learning theory in Section 2.5. In particular, we aim to understand how operational costs affect prediction (generalization) ability. This helps answer the third question Q3, about how intuition about operational cost can help produce a better probabilistic model. We show first that the hypothesis spaces for most of the applications in Section 2.3 can be bounded in a specific way - by an intersection of a ball and a halfspace - and this is true regardless of how complicated the constraints of the optimization problem are, and how different the operational costs are from each other in the different applications. Second, we bound the complexity of this type of hypothesis space using a technique based on Maurey's Lemma [Barron, 1993, Zhang, 2002] that leads eventually to a counting problem, where we calculate the number of integer points within a polyhedron in order to obtain a generalization bound. Our results show that it is possible to make use of much more general structure in estimation problems, compared to the standard (norm-constrained) structures like sparsity and smoothness; further, this additional structure can benefit generalization ability. A shorter version of this work has been previously published [see Tulabandhula and Rudin, 2012].

2.2 The Sequential and Simultaneous Processes

We have a training set of (random) labeled instances, $\{(x_i, y_i)\}_{i=1}^{n}$, where $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$ that we will use to learn a function $f^* : \mathcal{X} \to \mathcal{Y}$. Commonly in machine learning this is done by choosing f to be the solution of a minimization problem:

$$f^* \in \operatorname{argmin}_{f \in \mathcal{F}^{unc}} \left(\sum_{i=1}^{n} l(f(x_i), y_i) + C_2 R(f) \right), \tag{2.1}$$

for some loss function $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$, regularizer $R : \mathcal{F}^{unc} \to \mathbb{R}$, constant C_2 and function class $\mathcal{F}^{unc}$. Here, $\mathcal{Y} \subset \mathbb{R}$. Typical loss functions used in machine learning are the 0-1 loss, ramp loss, hinge loss, logistic loss and the exponential loss. Function class $\mathcal{F}^{unc}$ is commonly the class of all linear functionals, where an element $f \in \mathcal{F}^{unc}$ is of the form $\beta^T x$, where $\mathcal{X} \subset \mathbb{R}^p$, $\beta \in \mathbb{R}^p$. We have used '*unc*' in the superscript for $\mathcal{F}^{unc}$ to refer to the word "unconstrained," since it contains all linear functionals. Typical regularizers R are the ℓ_1 and ℓ_2 norms of β. Note that nonlinearities can be incorporated into $\mathcal{F}^{unc}$ by allowing nonlinear features, so that we now would have $f(x) = \sum_{j=1}^{p} \beta_j h_j(x)$, where $\{h_j\}_j$ is the set of features, which can be arbitrary nonlinear functions of x; for simplicity in notation, we will equate $h_j(x) = x_j$ and have $\mathcal{X} \subset \mathbb{R}^p$.

Consider an organization making policy decisions. Given a new collection of unlabeled instances $\{\tilde{x}_i\}_{i=1}^{m}$, the organization wants to create a policy π^* that minimizes a certain operational cost $\text{OpCost}(\pi, f^*, \{\tilde{x}_i\}_i)$. Of course, if the organization knew the true labels for the $\{\tilde{x}_i\}_i$'s beforehand, it would choose a policy to optimize the operational cost based directly on these labels, and would not need f^*. Since the labels are not known, the operational costs are calculated using the model's predictions, the $f^*(\tilde{x}_i)$'s. The difference between the traditional sequential process and the new simultaneous process is whether f^* is chosen with or without knowledge of the operational cost.

As an example, consider $\{\tilde{\mathbf{x}}_i\}_i$ as representing machines in a factory waiting to be repaired, where the first feature $\tilde{\mathbf{x}}_{i,1}$ is the age of the machine, the second feature $\tilde{\mathbf{x}}_{i,2}$ is the condition at its last inspection, etc. The value $f^*(\tilde{\mathbf{x}}_i)$ is the predicted probability of failure for $\tilde{\mathbf{x}}_i$. Policy π^* is the order in which the machines $\{\tilde{\mathbf{x}}_i\}_i$ are repaired, which is chosen based on how likely they are to fail, that is, $\{f^*(\tilde{\mathbf{x}}_i)\}_i$, and on the costs of the various types of repairs needed. The traditional sequential process picks a model f^*, based on past failure data without the knowledge of operational cost, and afterwards computes π^* based on an optimization problem involving the $\{f^*(\tilde{\mathbf{x}}_i)\}_i$'s and the operational cost. The new simultaneous process picks f^* and π^* at the same time, based on optimism or pessimism on the operational cost of π^*.

Formally, the **sequential process** computes the policy according to two steps, as follows.

Step 1: Create function f^* based on $\{(x_i, y_i)\}_i$ according to (2.1). That is

$$f^* \in \text{argmin}_{f\in\mathcal{F}unc} \left(\sum_{i=1}^{n} l(f(x_i), y_i) + C_2 R(f) \right).$$

Step 2: Choose policy π^* to minimize the operational cost,

$$\pi^* \in \text{argmin}_{\pi\in\Pi} \text{OpCost}(\pi, f^*, \{\tilde{x}_i\}_i).$$

The operational cost $\text{OpCost}(\pi, f^*, \{\tilde{x}_i\}_i)$ is the amount the organization will spend if policy π is chosen in response to the values of $\{f^*(\tilde{x}_i)\}_i$.

To define the **simultaneous process**, we combine Steps 1 and 2 of the sequential process. We can choose an **optimistic bias**, where we prefer (all else being equal) a model providing lower costs, or we can choose a **pessimistic bias** that prefers higher costs, where the degree of optimism or pessimism is controlled by a parameter C_1. in other words, the optimistic bias lowers costs when there is uncertainty, whereas the pessimistic bias raises them. The new steps are as follows.

Step 1: Choose a model f° obeying one of the following:

$$\text{Optimistic Bias: } f^\circ \in \operatorname*{argmin}_{f \in \mathcal{F}^{unc}} \Bigg[\sum_{i=1}^{n} l\left(f(x_i), y_i\right) + C_2 R(f) + C_1 \min_{\pi \in \Pi} \text{OpCost}\left(\pi, f, \{\tilde{x}_i\}_i\right) \Bigg] \tag{2.2}$$

$$\text{Pessimistic Bias: } f^\circ \in \operatorname*{argmin}_{f \in \mathcal{F}^{unc}} \Bigg[\sum_{i=1}^{n} l\left(f(x_i), y_i\right) + C_2 R(f) - C_1 \min_{\pi \in \Pi} \text{OpCost}\left(\pi, f, \{\tilde{x}_i\}_i\right) \Bigg]. \tag{2.3}$$

Step 2: Compute the policy:

$$\pi^\circ \in \operatorname*{argmin}_{\pi \in \Pi} \text{OpCost}\left(\pi, f^\circ, \{\tilde{x}_i\}_i\right).$$

When $C_1 = 0$, the simultaneous process becomes the sequential process; the sequential process is a special case of the simultaneous process.

The optimization problem in the simultaneous process can be computationally difficult, particularly if the subproblem to minimize OpCost involves discrete optimization. However, if the number of unlabeled instances is small, or if the policy decision can be broken into several smaller subproblems, then even if the training set is large, one can solve Step 1 using different types of mathematical programming solvers, including MINLP solvers [Bonami et al., 2008], Nelder-Mead [Nelder and Mead, 1965] and Alternating Minimization schemes [Tulabandhula et al., 2011]. One needs to be able to solve instances of that optimization problem in any case for Step 2 of the sequential process. The simultaneous process is more intensive than the sequential process in that it requires repeated solutions of that optimization problem, rather than a single solution.

The regularization term $R(f)$ can be for example, an ℓ_1 or ℓ_2 regularization term to encourage a sparse or smooth solution.

As the C_1 coefficient swings between large values for optimistic and pessimistic cases, the algorithm finds the best solution (having the lowest loss with respect to the data) for each possible cost. Once the regularization coefficient is too large, the algorithm will sacrifice empirical error in favor of lower costs, and will thus obtain solutions that are not reasonable. When that happens, we know we have already mapped out the full range of costs for reasonable solutions. This range can be used for pre-allocation decisions.

By sweeping over a range of C_1, we obtain a range of costs that we might incur. Based on this range, we can choose to allocate a reasonable amount of resources so that we can react best to whatever nature brings. This helps answer question Q1 in Section 2.1. In addition, we can pick a value of C_1 such that the resulting operational cost is a specific amount. In this case, we checking whether a probabilistic model exists, corresponding to that cost, that is reasonably supported by data. This can answer question Q2 in Section 2.1.

It is possible for the set of feasible policies Π to depend on recommendations $\{f(\tilde{x}_1), ..., f(\tilde{x}_m)\}$, so that $\Pi = \Pi(f, \{\tilde{x}_i\}_i)$ in general. We will revisit this possibility in Section 2.4. It is also possible for the optimization over $\pi \in \Pi$ to be trivial, or the optimization problem could have a closed form solution. Our notation does accommodate this, and is more general.

One should not view the operational cost as a utility function that needs to be estimated, as in reinforcement learning, where we do not know the cost. Here one knows precisely what the cost will be under each possible outcome. Unlike in reinforcement learning, we have a complicated one shot decision problem at hand and have training data as well as future/unlabeled examples on which the predictive model makes prediction on.

The use of unlabeled data $\{\tilde{x}_i\}_i$ has been explored widely in the machine learning literature under semi-supervised, transductive, and unsupervised learning. In particular, we point out that the simultaneous process is not a semi-supervised learning method [see Chapelle et al., 2006], since it does not use the unlabeled data to provide information about the underlying distribution. A small unlabeled sample is not very useful for semi-supervised learning, but could be very useful for constructing a low-cost policy. The simultaneous process also has a resemblance to transductive learning [see Zhu, 2007], whose goal is to produce the output labels on the set of unlabeled examples; in this case, we produce a function (namely the operational cost) applied to those output labels. The simultaneous process, for a fixed choice of C_1, can also be considered as a multi-objective machine learning method, since it involves an optimization problem having two terms with competing goals [see Jin, 2006].

2.2.1 The Simultaneous Process in the Context of Structural Risk Minimization

In the framework of statistical learning theory [e.g., Vapnik, 1998, Pollard, 1984, Anthony and Bartlett, 1999, Zhang, 2002], prediction ability of a class of models is guaranteed when the class has low "complexity," where complexity is defined via covering numbers, VC (Vapnik-Chervonenkis) dimension, Rademacher complexity, gaussian complexity, etc. Limiting the complexity of the hypothesis space imposes a bias, and the classical image associated with the bias-variance tradeoff is provided in Figure 2-1(a). The set of good models is indicated on the axis of the figure. Models that are not good are either overfitted (explaining too much of the variance of the data, having a high complexity), or underfitted (having too strong of a bias and a high empirical error). By understanding complexity, we can find a model class where both the training error and the complexity are kept low. An example of increasingly complex model classes is the set of nested classes of polynomials, starting with constants, then linear functions, second order polynomials and so on.

In predictive modeling problems, there is often no one right statistical model when dealing with finite datasets, in fact there may be a whole class of good models. In addition, it is possible that a small change in the choice of predictive model could lead to a large change in the cost required to implement the policy recommended by the model. This occurs, for instance, when costs are based on objects (e.g., products) that come in discrete amounts. Figure 2-1(b) illustrates this possibility, by showing that

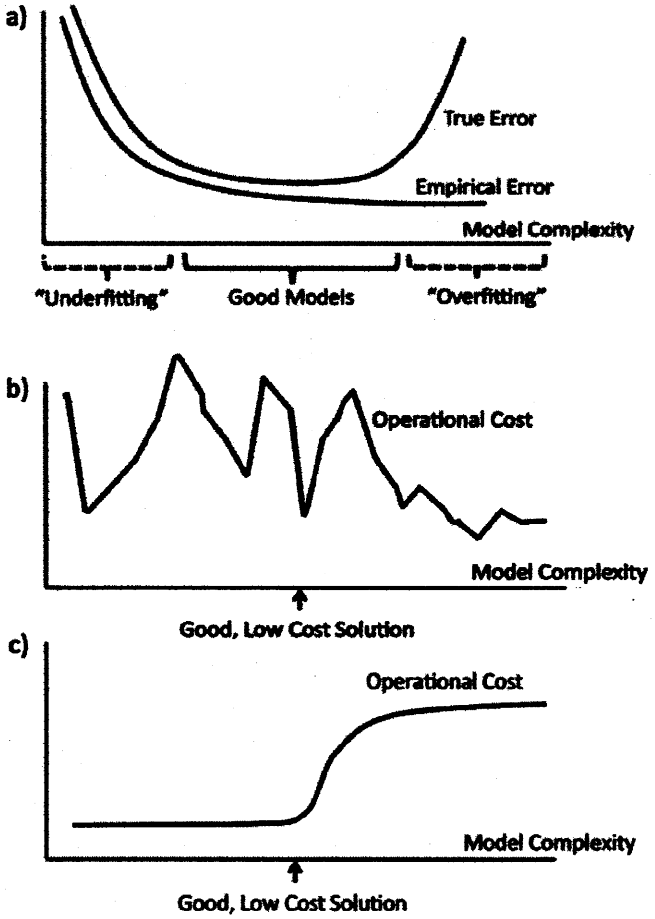


Figure 2-1: In all three plots, the x-axis represents model classes with increasing complexity. a) Relationship between training error and test error as a function of model complexity. b) A possible operational cost as a function of model complexity. c) Another possible operational cost.

there may be a variety of costs amongst the class of good models. The simultaneous process can find the range of costs for the set of good models, which can be used for allocation of costs, as discussed in the first question Q1 in the introduction.

Recall that question Q3 asked if our intuition about how much it will cost to solve a problem can help us produce a better probabilistic model. Figure 2-1 can be used to illustrate how this question can be answered. Assume we have a strong prior belief that the operational cost will not be above a certain fixed amount. Accordingly, we will choose only amongst the class of low cost models. This can significantly limit the complexity of the hypothesis space, because the set of low-cost good models might be much smaller than the full space of good models. Consider, for example, the cost displayed in Figure 2-1(c), where only models on the left part of the plot would be considered, since they are low cost models. Because the hypothesis space is smaller, we may be able to produce a tighter bound on the complexity of the hypothesis space, thereby obtaining a better prediction guarantee for the simultaneous process than for the sequential process. In Section 2.5 we develop results of this type. These results indicate that in some cases, the operational cost can be an important quantity for generalization.

2.3 Conceptual Demonstrations

We provide four examples. In the first, we estimate manpower requirements for a scheduling task. In the second, we estimate real estate prices for a purchasing decision. In the third, we estimate call arrival rates for a call center staffing problem. In the fourth, we estimate failure probabilities for manholes (access points to an underground electrical grid). The first two are small scale reproducible examples, designed to demonstrate new types of constraints due to operational costs. In the first example, the operational cost subproblem involves scheduling. In the second, it is a knapsack problem, and in the third, it is another multidimensional knapsack variant. In the fourth, it is a routing problem. In the first, second and fourth examples, the operational cost leads to a linear constraint, while in the third example, the cost leads to a quadratic constraint.

Throughout this section, we will assume that we are working with linear functions f of the form $\beta^T x$ so that $\Pi(f, \{\tilde{x}_i\}_i)$ can be denoted by $\Pi(\beta, \{\tilde{x}_i\}_i)$. We will set $R(f)$ to be equal to $\|\beta\|_2^2$. We will also use the notation $\mathcal{F}^R$ to denote the set of linear functions that satisfy an additional property:

$$\mathcal{F}^R := \{f \in \mathcal{F}^{unc} : R(f) \leq C_2^*\},$$

where C_2^* is a known constant greater than zero. We will use constant C_2 from (2.1), and also C_2^* from the definition of $\mathcal{F}^R$, to control the extent of regularization. C_2 is inversely related to C_2^*. We use both versions interchangeably throughout the chapter.

2.3.1 Manpower Data and Scheduling with Precedence Constraints

We aim to schedule the starting times of medical staff, who work at 6 stations, for instance, ultrasound, X-ray, MRI, CT scan, nuclear imaging, and blood lab. Current and incoming patients need to go through some of these stations in a particular order. The six stations and the possible orders are shown in Figure 2-2. Each station is denoted by a line. Work starts at the check-in (at time π_1) and ends at the check-out (at time π_5). The stations are numbered 6-11, in order to avoid confusion with the times π_1-π_5. The clinic has precedence constraints, where a station cannot be staffed (or work with patients) until the preceding stations are likely to finish with their patients. For instance, the check-out should not start until all the previous stations finish. Also, as shown in Figure 2-2, station 11 should not start until stations 8 and 9 are complete at time π_4, and station 9 should not start until station 7 is complete at time π_3. Stations 8 and 10 should not start until station 6 is complete. (This is related to a similar problem called *planning with preference* posed by F. Malucelli, Politecnico di Milano).

The operational goal is to minimize the total time of the clinic's operation, from when the check-in happens at time π_1 until the check-out happens at time π_5. We

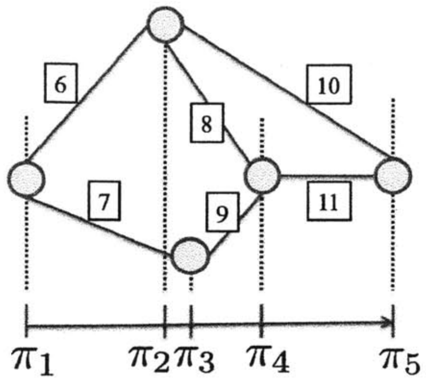


Figure 2-2: Staffing estimation with bias on scheduling with precedence constraints.

estimate the time it takes for each station to finish its job with the patients based on two variables: the new load of patients for the day at the station, and the number of current patients already present. The data are available as *manpower* in the R-package *bestglm*, using "Hour," "Load" and "Stay" columns. The training error is chosen to be the least squares loss between the estimated time for stations to finish their jobs ($\beta^T x_i$) and the actual times it took to finish (y_i). The unlabeled data are the new load and current patients present at each station for a given period, given as $\tilde{x}_6, .., \tilde{x}_{11}$. Let π denote the 5-dimensional real vector with coordinates $\pi_1, ..., \pi_5$.

The operational cost is the total time $\pi_5 - \pi_1$. Step 1, with an optimistic bias, can be written as:

$$\min_{\{\beta:\|\beta\|_2^2 \le C_2^*\}} \sum_{i=1}^{n} (y_i - \beta^T x_i)^2 + C_1 \min_{\pi \in \Pi(\beta, \{\tilde{x}_i\}_i)} (\pi_5 - \pi_1), \tag{2.4}$$

where the feasible set $\Pi(\beta, \{\tilde{x}_i\}_i)$ is defined by the following constraints:

$$\pi_a + \beta^T \tilde{x}_i \le \pi_b; \;\; (a, i, b) \in \{(1,6,2), (1,7,3), (2,8,4), (3,9,4), (2,10,5), (4,11,5)\}$$
$$\pi_a \ge 0 \text{ for } a = 1, ..., 5.$$

To solve (2.4) given values of C_1 and C_2, we used a function-evaluation-based scheme called Nelder-Mead [Nelder and Mead, 1965] where at every iterate of β, the sub-

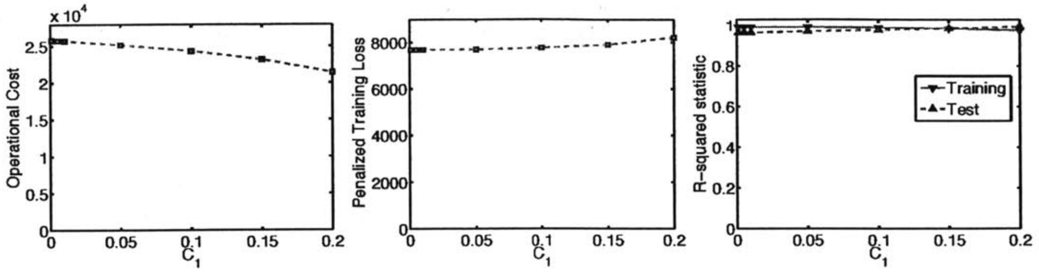


Figure 2-3: *Left:* Operational cost vs C_1. *Center:* Penalized training loss vs C_1. *Right:* R-squared statistic. $C_1 = 0$ corresponds to the baseline, which is the sequential formulation.

problem for π was solved to optimality (using Gurobi[1]). C_2 was chosen heuristically based on (2.1) and kept fixed for the experiment beforehand.

Figure 2-3 shows the operational cost, training loss, and r^2 statistic[2] for various values of C_1. For C_1 values between 0 and 0.2, the operational cost varies substantially, by ~16%. The r^2 values for both training and test vary much less, by ~3.5%, where the best value happened to have the largest value of C_1. For small datasets, there is generally a variation between training and test: for this data split, there is a 3.16% difference in r^2 between training and test for plain least squares, and this is similar across various splits of the training and test data. This means that for the scheduling problem, there is a range of reasonable predictive models within about ~3.5% of each other.

What we learn from this, in terms of the three questions in the introduction, is that: 1) There is a wide range of possible costs within the range of reasonable optimistic models. 2) We have found a reasonable scenario, supported by data, where the cost is 16% lower than in the sequential case. 3) If we have a prior belief that the cost will be lower, the models that are more accurate are the ones with lower costs, and therefore we may not want to designate the full cost suggested by the sequential process. We can perhaps designate up to 16% less.

Connection to learning theory: In the experiment, we used tradeoff parameter C_1

[1] Gurobi Optimizer v3.0, Gurobi Optimization, Inc. 2010.

[2] If $\hat{y}_i$ are the predicted labels and $\bar{y}$ is the mean of $\{y_1, ..., y_n\}$, then the value of the r^2 statistic is defined as $1 - \sum_i (y_i - \hat{y}_i)^2 / \sum_i (y_i - \bar{y})^2$. Thus r^2 is an affine transformation of the sum of squares error. r^2 allows training and test accuracy to be measured on a comparable scale.

to provide a soft constraint. Considering instead the corresponding hard constraint $\min_\pi(\pi_5 - \pi_1) \leq \alpha$, the total time must be at least the time for any of the three paths in Figure 2-2, and thus at least the average of them:

$$\begin{aligned}\alpha \geq &\min_{\pi \in \Pi\{\beta, \{\tilde{x}_i\}_i\}} \pi_5 - \pi_1 \\ \geq &\max\{(\tilde{x}_6 + \tilde{x}_{10})^T\beta, (\tilde{x}_6 + \tilde{x}_8 + \tilde{x}_{11})^T\beta, (\tilde{x}_7 + \tilde{x}_9 + \tilde{x}_{11})^T\beta\} \\ \geq &z^T\beta \end{aligned} \tag{2.5}$$

where

$$z = \frac{1}{3}[(\tilde{x}_6 + \tilde{x}_{10}) + (\tilde{x}_6 + \tilde{x}_8 + \tilde{x}_{11}) + (\tilde{x}_7 + \tilde{x}_9 + \tilde{x}_{11})].$$

The main result in Section 2.5, Theorem 2.5.1, is a learning theoretic guarantee in the presence of this kind of arbitrary linear constraint, $z^T\beta \leq \alpha$.

2.3.2 Housing Prices and the Knapsack Problem

A developer will purchase 3 properties amongst the 6 that are currently for sale and in addition, will remodel them. She wants to maximize the total value of the houses she picks (the value of a property is its purchase cost plus the fixed remodeling cost). The fixed remodeling costs for the 6 properties are denoted $\{c_i\}_{i=1}^6$. She estimates the purchase cost of each property from data regarding historical sales, in this case, from the *Boston Housing* data set [Bache and Lichman, 2013], which has 13 features. Let policy $\pi \in \{0,1\}^6$ be the 6-dimensional binary vector that indicates the properties she purchases. Also, x_i represents the features of property i in the training data and $\tilde{x}_i$ represents the features of a different property that is currently on sale. The training loss is chosen to be the sum of squares error between the estimated prices $\beta^T x_i$ and the true house prices y_i for historical sales. The cost (in this case, total value) is the sum of the three property values plus the costs for repair work. A pessimistic bias on total value is chosen to motivate a min-max formulation. The resulting (mixed-integer) program for Step 1 of the simultaneous process is:

$$\min_{\beta\in\{\beta:\beta\in\mathbf{R}^{13},\|\beta\|_2^2\le C_2^*\}} \sum_{i=1}^{n}(y_i-\beta^T x_i)^2$$
$$+C_1\left[\max_{\pi\in\{0,1\}^6}\sum_{i=1}^{6}(\beta^T\tilde{x}_i+c_i)\pi_i \quad \text{subject to} \quad \sum_{i=1}^{6}\pi_i\le 3\right]. \tag{2.6}$$

Notice that the second term above is a 1-dimensional $\{0,1\}$ knapsack instance. Since the set of policies Π does not depend on β, we can rewrite (2.6) in a cleaner way that was not possible directly with (2.4):

$$\min_{\beta}\max_{\pi}\left[\sum_{i=1}^{n}(y_i-\beta^T x_i)^2+C_1\sum_{i=1}^{6}(\beta^T\tilde{x}_i+c_i)\pi_i\right]$$

subject to

$$\beta\in\{\beta:\beta\in\mathbf{R}^{13},\|\beta\|_2^2\le C_2^*\}$$
$$\pi\in\left\{\pi:\pi\in\{0,1\}^6,\sum_{i=1}^{6}\pi_i\le 3\right\}. \tag{2.7}$$

To solve (2.7) with user-defined parameters C_1 and C_2, we use fminimax, available through Matlab's Optimization toolbox.[3]

For the training and unlabeled set we chose, there is a change in policy above and below $C_1 = 0.05$, where different properties are purchased. Figure 2-4 shows the operational cost which is the predicted total value of the houses after remodeling, the training loss, and r^2 values for a range of C_1. The training loss and r^2 values change by less than ~3.5%, whereas the total value changes about 6.5%. We can again draw conclusions in terms of the questions in the introduction as follows. The pessimistic bias shows that even if the developer chose the best response policy to the prices, she might end up with the expected total value of the purchased properties on the order of 6.5% less if she is unlucky. Also, we can now produce a realistic model where the total value is 6.5% less. We can use this model to help her understand the uncertainty involved in her investment.

[3] Version 5.1, Matlab R2010b, Mathworks, Inc.

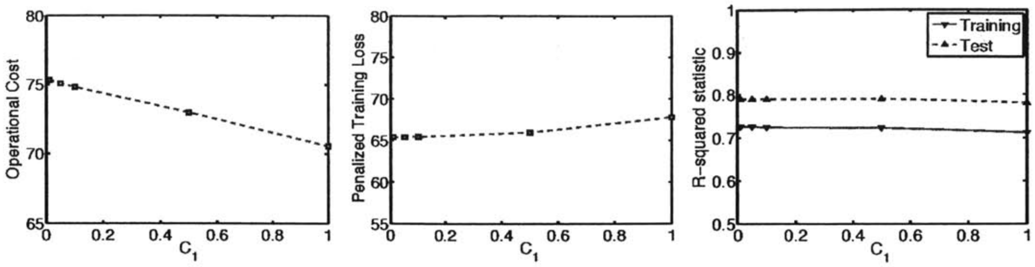


Figure 2-4: *Left:* Operational cost (total value) vs C_1. *Center:* Penalized training loss vs C_1. *Right:* R-squared statistic. $C_1 = 0$ corresponds to the baseline, which is the sequential formulation.

Before moving to the next application of the proposed framework, we provide a bound analogous to that of (2.5). Let us replace the soft constraint represented by the second term of (2.6) with a hard constraint and then obtain a lower bound:

$$\alpha \geq \max_{\pi \in \{0,1\}^6, \sum_{i=1}^{6} \pi_i \leq 3} \sum_{i=1}^{6} (\beta^T \tilde{x}_i)\pi_i \geq \sum_{i=1}^{6} (\beta^T \tilde{x}_i)\pi'_i, \tag{2.8}$$

where π' is some feasible solution of the linear programming relaxation of this problem that also gives a lower objective value. For instance picking $\pi'_i = 0.5$ for $i = 1, \ldots, 6$ is a valid lower bound giving us a looser constraint. The constraint can be rewritten:

$$\beta^T \left(\frac{1}{2} \sum_{i=1}^{n} \tilde{x}_i \right) \leq \alpha.$$

This is again a linear constraint on the function class parametrized by β, which we can use for the analysis in Section 2.5.

Note that if all six properties were being purchased by the developer instead of three, the knapsack problem would have a trivial solution and the regularization term would be explicit (rather than implicit).

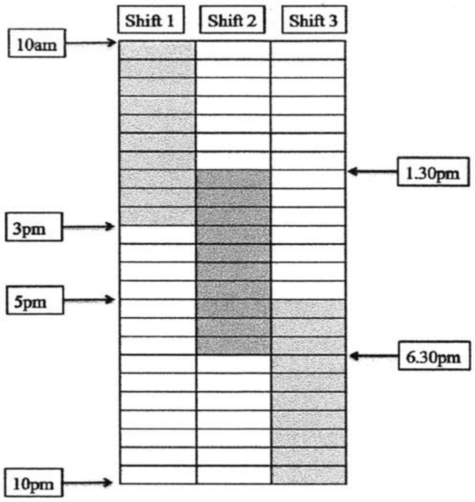


Figure 2-5: The three shifts for the call center. The cells represent half-hour periods, and there are 24 periods per work day. Work starts at 10am and ends at 10pm.

2.3.3 A Call Center's Workload Estimation and Staff Scheduling

A call center management wants to come up with the per-half-hour schedule for the staff for a given day between 10am to 10pm. The staff on duty should be enough to meet the demand based on call arrival estimates $N(i), i = 1, ..., 24$. The staff required will depend linearly on the demand per half-hour. The demand per half-hour in turn will be computed based on the Erlang C model [Aldor-Noiman et al., 2009] which is also known as the square-root staffing rule. This particular model relates the demand $D(i)$ to the call arrival rate $N(i)$ in the following manner: $D(i) \propto N(i) + c\sqrt{N(i)}$ where c determines where on the QED (Quality Efficiency Driven) curve the center wants to operate on. We make the simplifying assumptions that the service time for each customer is constant, and that the coefficient c is 0.

If we know the call arrival rate $N(i)$, we can calculate the staffing requirements during each half hour. If we do not know the call arrival rate, we can estimate it from past data, and make optimistic or pessimistic staffing allocations.

There are additional staffing constraints as shown in Figure 2-5, namely, there are three sets of employees who work at the center such that: the first set can work only from 10am-3pm, the second can work from 1:30pm-6:30pm, and the third set works from 5pm-10pm. The operational cost is the total number of employees hired to work that day (times a constant, which is the amount each person is paid). The objective of the management is to reduce the number of staff on duty but at the same time maintain a certain quality and efficiency.

The call arrivals are modeled as a poisson process [Aldor-Noiman et al., 2009]. What previous studies [Brown et al., 2001] have discovered about this estimation problem is that the square root of the call arrival rate tends to behave as a linear function of several features, including: day of the week, time of the day, whether it is a holiday/irregular day, and whether it is close to the end of the billing cycle.

Data for call arrivals and features were collected over a period of 10 months from Mid-February 2004 to the end of December 2004 [this is the same dataset as in Aldor-Noiman et al., 2009]. After converting categorical variables into binary encodings (e.g., each of the 7 weekdays into 6 binary features) the number of features is 36, and we randomly split the data into a training set and test set (2764 instances for training; another 3308 for test).

We now formalize the optimization problem for the simultaneous process. Let policy $\pi \in \mathbb{Z}_+^3$ be a size three vector indicating the number of employees for each of the three shifts. The training loss is the sum of squares error between the estimated square root of the arrival rate $\beta^T x_i$ and the actual square root of the arrival rate $y_i := \sqrt{N(i)}$. The cost is proportional to the total number of employees signed up to work, $\sum_i \pi_i$. An optimistic bias on cost is chosen, so that the (mixed-integer) program for Step 1 is:

$$\min_{\beta:\|\beta\|_2^2 \leq C_2^*} \sum_{i=1}^{n} (y_i - \beta^T x_i)^2$$
$$+C_1 \left[\min_{\pi} \sum_{i=1}^{3} \pi_i \text{ subject to } a_i^T \pi \geq (\beta^T \tilde{x}_i)^2 \text{ for } i = 1, ..., 24, \pi \in \mathbb{Z}_+^3 \right], \quad (2.9)$$

where Figure 2-5 illustrates the matrix A with the shaded cells containing entry 1

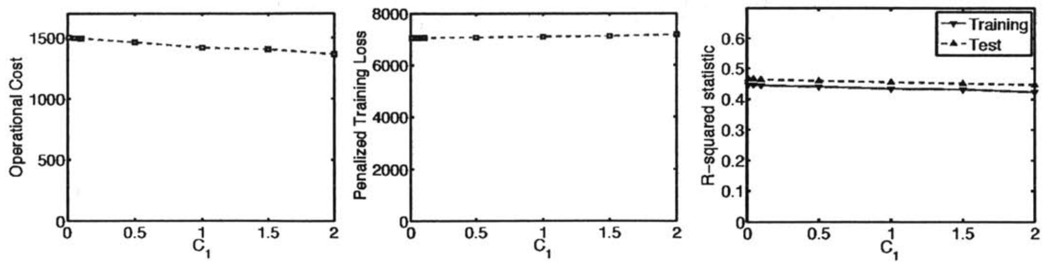


Figure 2-6: *Left:* Operational cost vs C_1. *Center:* Penalized training loss vs C_1. *Right:* R-squared statistic. $C_1 = 0$ corresponds to the baseline, which is the sequential formulation.

and 0 elsewhere. The notation a_i indicates the i^{th} row of A:

$$a_i(j) = \begin{cases} 1 & \text{if staff } j \text{ can work in half-hour period } i \\ 0 & \text{otherwise.} \end{cases}$$

To solve (2.9) we first relax the ℓ_2-norm constraint on β by adding another term to the function evaluation, namely $C_2\|\beta\|_2^2$. This, way we can use a function-evaluation based scheme that works for unconstrained optimization problems. As in the manpower scheduling example, we used an implementation of the Nelder-Mead algorithm, where at each step, Gurobi was used to solve the mixed-integer subproblem for finding the policy.

Figure 2-6 shows the operational cost, the training loss, and r^2 values for a range of C_1. The training loss and r^2 values change only ~1.6% and ~3.9% respectively, whereas the operational cost changes about 9.2%. Similar to the previous two examples, we can again draw conclusions in terms of the questions in Section 2.1 as follows. The optimistic bias shows that the management might incur operational costs on the order of 9% less if they are lucky. Further, the simultaneous process produces a reasonable model where costs are about 9% less. If the management team believes they will be reasonably lucky, they can justify designating substantially less than the amount suggested by the traditional sequential process.

Let us now investigate the structure of the operational cost regularization term we have in (2.9). For convenience, let us stack the quantities $(\beta^T \tilde{x}_i)^2$ as a vector

$b \in \mathbb{R}^{24}$. Also let boldface symbol $\mathbf{1}$ represent a vector of all ones. If we replace the soft constraint represented by the second term with a hard constraint having an upper bound α, we get:

$$\alpha \geq \min_{\pi \in \mathbf{Z}_+^3; A\pi \geq b} \sum_{i=1}^{3} \mathbf{1}^T \pi \overset{(\dagger)}{\geq} \min_{\pi \in \mathbf{R}_+^3; A\pi \geq b} \sum_{i=1}^{3} \mathbf{1}^T \pi \overset{(\ddagger)}{=} \max_{w \in \mathbf{R}_+^{24}; A^T w \leq \mathbf{1}} \sum_{i=1}^{24} w_i (\beta^T \tilde{x}_i)^2$$
$$\overset{(*)}{\geq} \sum_{i=1}^{24} \frac{1}{10} (\beta^T \tilde{x}_i)^2.$$

Here α is related to the choice of C_1 and is fixed. (†) represents an LP relaxation of the integer program with π now belonging to the positive orthant rather than the cartesian product of set of positive integers. (‡) is due to LP strong duality and ($*$) is by choosing an appropriate feasible dual variable. Specifically, we pick $w_i = \frac{1}{10}$ for $i = 1, \ldots, 24$, which is feasible because staff cannot work more than 10 half hour shifts (or 5 hours). With the three inequalities, we now have a constraint on β of the form:

$$\sum_{i=1}^{24} (\beta^T \tilde{x}_i)^2 \leq 10\alpha.$$

This is a quadratic form in β and gives an ellipsoidal feasible set. We already had a simple ellipsoidal feasibility constraint while defining the minimization problem of (2.9) of the form $\|\beta\|_2^2 \leq C_2^*$. Thus, we can see that our effective hypothesis set (the set of linear functionals satisfying these constraints) has become smaller. This in turn affects generalization. We are investigating generalization bounds for this type of hypothesis set in separate ongoing work.

2.3.4 The Machine Learning and Traveling Repairman Problem (ML&TRP) [Tulabandhula et al., 2011]

Recently, power companies have been investing in intelligent "proactive" maintenance for the power grid, in order to enhance public safety and reliability of electrical service. For instance, New York City has implemented new inspection and repair programs for manholes, where a manhole is an access point to the underground electrical system.

Electrical grids can be extremely large (there are on the order of 23,000-53,000 manholes in each borough of NYC), and parts of the underground distribution network in many cities can be as old as 130 years, dating from the time of Thomas Edison. Because of the difficulties in collecting and analyzing historical electrical grid data, electrical grid repair and maintenance has been performed reactively (fix it only when it breaks), until recently [Urbina, 2004]. These new proactive maintenance programs open the door for machine learning to assist with smart grid maintenance.

Machine learning models have started to be used for proactive maintenance in NYC, where supervised ranking algorithms are used to rank the manholes in order of predicted susceptibility to failure (fires, explosions, smoke) so that the most vulnerable manholes can be prioritized [Rudin et al., 2010, 2012a, 2011]. The machine learning algorithms make reasonably accurate predictions of manhole vulnerability; however, they do not (nor would they, using any other prediction-only technique) take the cost of repairs into account when making the ranked lists. They do not know that it is unreasonable, for example, if a repair crew has to travel across the city and back again for each manhole inspection, losing important time in the process. The power company must solve an optimization problem to determine the best repair route, based on the machine learning model's output. We might wish to find a policy that is not only supported by the historical power grid data (that ranks more vulnerable manholes above less vulnerable ones), but also would give a better route for the repair crew. An algorithm that could find such a route would lead to an improvement in repair operations on NYC's power grid, other power grids across the world, and improvements in many different kinds of routing operations (delivery trucks, trains, airplanes).

The simultaneous process could be used to solve this problem, where the operational cost is the price to route the repair crew along a graph, and the probabilities of failure at each node in the graph must be estimated. We call this the "the machine learning and traveling repairman problem" (ML&TRP) and in our ongoing work [Tulabandhula et al., 2011] , we have developed several formulations for the ML&TRP. We demonstrated, using manholes from the Bronx region of NYC, that it is possible to obtain a much more practical route using the ML&TRP, by taking the cost of the route optimistically into account in the machine learning model. We showed also that from the routing problem, we can obtain a linear constraint on the hypothesis space, in order to apply the generalization analysis of Section 2.5 (and in order to address question Q3 of Section 2.1).

2.4 Connections to Robust Optimization

The goal of robust optimization (RO) is to provide the best possible policy that is acceptable under a wide range of situations.[4] This is different from the simultaneous process, which aims to find the best policies and costs for specific situations. Note that it is not always desirable to have a policy that is robust to a wide range of situations; this is a question of whether to respond to every situation simultaneously or whether to understand the single worst situation that could reasonably occur (which is what the pessimistic simultaneous formulation handles). In general, robust optimization can be overly pessimistic, requiring us to allocate enough to handle all reasonable situations; it can be substantially more pessimistic than the pessimistic simultaneous process.

In robust optimization, if there are several real-valued parameters involved in the optimization problem, we might declare a reasonable range, called the "uncertainty set," for each parameter (*e.g.* $a_1 \in [9, 10]$, $a_2 \in [1, 2]$). Using techniques of RO, we would minimize the largest possible operational cost that could arise from parameter settings in these ranges. Estimation is not usually involved in the study of robust optimization [with some exceptions, see Xu et al., 2009, who consider support vector machines]. On the other hand, one could choose the uncertainty set according to a statistical model, which is how we will build a connection to RO. Here, we choose the uncertainty set to be the class of models that fit the data to within ϵ, according to some fitting criteria.

The major goals of the field of RO include algorithms, geometry, and tractability in

[4] http://en.wikipedia.org/wiki/Robust_optimization

finding the best policy, whereas our work is not concerned with finding a robust policy, but we are concerned with estimation, taking the policy into account. Tractability for us is not always a main concern as we need to be able to solve the optimization problem, even to use the sequential process. Using even a small optimization problem as the operational cost might have a large impact on the model and decision. If the unlabeled set is not too large, or if the policy optimization problem can be broken into smaller subproblems, there is no problem with tractability. An example where the policy optimization might be broken into smaller subproblems is when the policy involves routing several different vehicles, where each vehicle must visit part of the unlabeled set; in that case there is a small subproblem for each vehicle. On the other hand, even though the goals of the simultaneous process and RO are entirely different, there is a strong connection with respect to the formulations for the simultaneous process and RO, and a class of problems for which they are equivalent. We will explore this connection in this section.

There are other methods that consider uncertainty in optimization, though not via the lens of estimation and learning. In the simplest case, one can perform both local and global sensitivity analysis for linear programs to ascertain uncertainty in the optimal solution and objective, but these techniques generally only handle simple forms of uncertainty [Vanderbei, 2008]. Our work is also related to stochastic programming, where the goal is to find a policy that is robust to almost all of the possible circumstances (rather than all of them), where there are random variables governing the parameters of the problem, with known distributions [Birge and Louveaux, 1997]. Again, our goal is not to find a policy that is necessarily robust to (almost all of) the worst cases, and estimation is again not the primary concern for stochastic programming, rather it is how to take known randomness into account when determining the policy.

2.4.1 Equivalence Between RO and the Simultaneous Process in Some Cases

In this subsection we will formally introduce RO. In order to connect RO to estimation, we will define the uncertainty set for RO, denoted $\mathcal{F}_{good}$, to be models for which the average loss on the sample is within ϵ of the lowest possible. Then we will present the equivalence relationship between RO and the simultaneous process, using a minimax theorem.

In Section 2.2, we had introduced the notation $\{(x_i, y_i)\}_i$ and $\{\tilde{x}_i\}_i$ for labeled and unlabeled data respectively. We had also introduced the class $\mathcal{F}^{unc}$ in which we were searching for a function f^* by minimizing an objective of the form (2.1). The uncertainty set $\mathcal{F}_{good}$ will turn out to be a subset of $\mathcal{F}^{unc}$ that depends on $\{(x_i, y_i)\}_i$ and f^* but not on $\{\tilde{x}_i\}_i$.

We start with plain (non-robust) optimization, using a general version of the vanilla sequential process. Let f denote an element of the set $\mathcal{F}_{good}$, where f is predetermined, known and fixed. Let the optimization problem for the policy decision π be defined by:

$$\min_{\pi \in \Pi(f;\{\tilde{x}\}_i)} \text{OpCost}(\pi, f; \{\tilde{x}_i\}), \qquad \textit{(Base problem)} \qquad (2.10)$$

where $\Pi(f; \{\tilde{x}_i\})$ is the feasible set for the optimization problem. Note that this is a more general version of the sequential process than in Section 2.2, since we have allowed the constraint set Π to be a function of both f and $\{\tilde{x}_i\}_i$, whereas in (2.2) and (2.3), only the objective and not the constraint set can depend on f and $\{\tilde{x}_i\}_i$. Allowing this more general version of Π will allow us to relate (2.10) to RO more clearly, and will help us to specify the additional assumptions we need in order to show the equivalence relationship. Specifically, in Section 2.2, OpCost depends on $(f, \{\tilde{x}_i\}_i)$ but not Π; whereas in RO, generally Π depends on $(f, \{\tilde{x}_i\}_i)$ but not OpCost. The fact that OpCost does not need to depend on f and $\{\tilde{x}_i\}_i$ is not a serious issue, since we can generally remove their dependence through auxiliary variables. For instance, if the problem is a minimization of the form (2.10), we can use an auxiliary variable, say t, to obtain an equivalent problem:

$$\begin{aligned} &\min_{\pi,t} t \qquad \textit{(Base problem reformulated)} \\ \text{such that } &\pi \in \Pi(f; \{\tilde{x}_i\}) \\ &\text{OpCost}(\pi, f; \{\tilde{x}_i\}) \leq t \end{aligned}$$

where the dependence on $(f, \{\tilde{x}_i\}_i)$ is present only in the (new) feasible set. Since we had assumed f to be fixed, this is a deterministic optimization problem (convex, mixed-integer, nonlinear, etc.).

Now, consider the case when f is not known exactly but only known to lie in the uncertainty set $\mathcal{F}_{good}$. The robust counterpart to (2.10) can then be written as:

$$\min_{\pi \in \bigcap_{g \in \mathcal{F}_{good}} \Pi(g; \{\tilde{x}\}_i)} \max_{f \in \mathcal{F}_{good}} \text{OpCost}(\pi, f; \{\tilde{x}_i\}) \qquad \textit{(Robust counterpart)} \quad (2.11)$$

where we obtain a "robustly feasible solution" that is guaranteed to remain feasible for all values of $f \in \mathcal{F}_{good}$. In general, (2.11) is much harder to solve than (2.10) and is a topic of much interest in the robust optimization community. As we discussed earlier, there is no focus in (2.11) on estimation, but it is possible to embed an estimation problem within the description of the set $\mathcal{F}_{good}$, which we now define formally.

In Section 2.3, $\mathcal{F}^R$ (a subset of $\mathcal{F}^{unc}$) was defined as the set of linear functionals with the property that $R(f) \leq C_2^*$. That is,

$$\mathcal{F}^R = \{f : f \in \mathcal{F}^{unc}, R(f) \leq C_2^*\}.$$

We define $\mathcal{F}_{good}$ as a subset of $\mathcal{F}^R$ by adding an additional property:

$$\mathcal{F}_{good} = \left\{ f : f \in \mathcal{F}^R, \sum_{i=1}^{n} l\left(f(x_i), y_i\right) \leq \sum_{i=1}^{n} l\left(f^*(x_i), y_i\right) + \epsilon \right\}, \qquad (2.12)$$

for some fixed positive real ϵ. In (2.12), again f^* is a solution that minimizes the objective in (2.1) over $\mathcal{F}^{unc}$. The right hand side of the inequality in (2.12) is thus constant, and we will henceforth denote it with a single quantity C_1^*. Substituting this definition of $\mathcal{F}_{good}$ in (2.11), and further making an important assumption (denoted **A1**) that Π is not a function of $(f, \{\tilde{x}_i\}_i)$, we get the following optimization problem:

$$\min_{\pi\in\Pi} \max_{\{f\in\mathcal{F}^R:\sum_{i=1}^n l(f(x_i),y_i)\le C_1^*\}} \left[\text{OpCost}\left(\pi, f, \{\tilde{x}_i\}_i\right)\right] \quad \textit{(Robust counterpart with assumptions)} \tag{2.13}$$

where C_1^* now controls the amount of the uncertainty via the set $\mathcal{F}_{good}$.

Before we state the equivalence relationship, we restate the formulations for optimistic and pessimistic biases on operational cost in the simultaneous process from (2.2) and (2.3):

$$\begin{aligned} &\min_{f\in\mathcal{F}^{unc}} \left[\sum_{i=1}^n l\left(f(x_i), y_i\right) + C_2 R(f) + C_1 \min_{\pi\in\Pi} \text{OpCost}\left(\pi, f, \{\tilde{x}_i\}_i\right)\right] \textit{(Simultaneous optimistic)} \\ &\min_{f\in\mathcal{F}^{unc}} \left[\sum_{i=1}^n l\left(f(x_i), y_i\right) + C_2 R(f) - C_1 \min_{\pi\in\Pi} \text{OpCost}\left(\pi, f, \{\tilde{x}_i\}_i\right)\right] \textit{(Simultaneous pessimistic)} \end{aligned} \tag{2.14}$$

Apart from the assumption **A1** on the decision set Π that we made in (2.13), we will also assume that $\mathcal{F}_{good}$ defined in (2.12) is convex; this will be assumption **A2**. If we also assume that the objective OpCost satisfies some nice properties (**A3**), and that uncertainty is characterized via the set $\mathcal{F}_{good}$, then we can show that the two problems, namely (2.14) and (2.13), are equivalent. Let $\Leftrightarrow$ denote equivalence between two problems, meaning that a solution to one side translates into the solution of the other side for some parameter values (C_1, C_1^*, C_2, C_2^*).

Proposition 2.4.1. Let $\Pi(f; \{\tilde{x}_i\}_i) = \Pi$ be compact, convex, and independent of parameters f and $\{\tilde{x}_i\}_i$ (assumption **A1**). Let $\{f \in \mathcal{F}^R : \sum_{i=1}^n l(f(x_i), y_i) \le C_1^*\}$ be convex (assumption **A2**). Let the cost (to be minimized) $\text{OpCost}(\pi, f, \{\tilde{x}_i\}_i)$ be concave continuous in f and convex continuous in π (assumption **A3**). Then, the robust optimization problem (2.13) is equivalent to the pessimistic bias optimization problem (2.14). That is,

$$\min_{\pi\in\Pi} \max_{\{f\in\mathcal{F}^R:\sum_{i=1}^n l(f(x_i),y_i)\leq C_1^*\}} \left[\mathrm{OpCost}(\pi, f, \{\tilde{x}_i\}_i)\right] \Leftrightarrow$$
$$\min_{f\in\mathcal{F}^{unc}} \left[\sum_{i=1}^n l\left(f(x_i), y_i\right) + C_2R(f) - C_1 \min_{\pi\in\Pi} \mathrm{OpCost}\left(\pi, f, \{\tilde{x}_i\}_i\right)\right].$$

Remark 2.4.2. That the equivalence applies to linear programs (LPs) is clear because the objective is linear and the feasible set is generally a polyhedron, and is thus convex. For integer programs, the objective OpCost satisfies continuity, but the feasible set is typically not convex, and hence, the result does not generally apply to integer programs. In other words, the requirement that the constraint set Π be convex excludes integer programs.

To prove Proposition 2.4.1, we restate a well-known generalization of von Neumann's minimax theorem and some related definitions.

Definition 1. A linear topological space (also called a topological vector space) is a vector space over a topological field (typically, the real numbers with their standard topology) with a topology such that vector addition and scalar multiplication are continuous functions. For example, any normed vector space is a linear topological space. A function h is upper semicontinuous at a point p_0 if for every $\epsilon > 0$ there exists a neighborhood U of p_0 such that $h(p) \leq h(p_0) + \epsilon$ for all $p \in U$. A function h defined over a convex set is quasi-concave if for all p, q and $\lambda \in [0, 1]$ we have $h(\lambda p + (1 - \lambda)q) \geq \min(h(p), h(q))$. Similar definitions follow for lower semicontinuity and quasi-convexity.

Theorem 2.4.3. [Sion's minimax theorem Sion, 1958] Let Π be a compact convex subset of a linear topological space and Ξ be a convex subset of a linear topological space. Let $G(\pi, \xi)$ be a real function on $\Pi \times \Xi$ such that

(i) $G(\pi, \cdot)$ is upper semicontinuous and quasi-concave on Ξ for each $\pi \in \Pi$;

(ii) $G(\cdot, \xi)$ is lower semicontinuous and quasi-convex on Π for each $\xi \in \Xi$.

Then

$$\min_{\pi \in \Pi} \sup_{\xi \in \Xi} G(\pi, \xi) = \sup_{\xi \in \Xi} \min_{\pi \in \Pi} G(\pi, \xi)$$

We can now proceed to the proof of Proposition (2.4.1).

Proof. (Of Proposition 2.4.1) We start from the left hand side of the equivalence we want to prove:

$$\begin{aligned}
&\min_{\pi \in \Pi} \max_{\{f \in \mathcal{F}^R : \sum_{i=1}^n l(f(x_i), y_i) \leq C_1^*\}} \left[\mathrm{OpCost}(\pi, f, \{\tilde{x}_i\}_i) \right] \\
\overset{(a)}{\Leftrightarrow} &\max_{\{f \in \mathcal{F}^R : \sum_{i=1}^n l(f(x_i), y_i) \leq C_1^*\}} \min_{\pi \in \Pi} \left[\mathrm{OpCost}(\pi, f, \{\tilde{x}_i\}_i) \right] \\
\overset{(b)}{\Leftrightarrow} &\max_{f \in \mathcal{F}^{unc}} \left[-\frac{1}{C_1} \left(\sum_{i=1}^n l(f(x_i), y_i) - C_1^* \right) - \frac{C_2}{C_1} \left(R(f) - C_2^* \right) + \min_{\pi \in \Pi} \mathrm{OpCost}(\pi, f, \{\tilde{x}_i\}_i) \right] \\
\overset{(c)}{\Leftrightarrow} &\min_{f \in \mathcal{F}^{unc}} \left[\sum_{i=1}^n l\left(f(x_i), y_i\right) + C_2 R(f) - C_1 \min_{\pi \in \Pi} \mathrm{OpCost}\left(\pi, f, \{\tilde{x}_i\}_i\right) \right].
\end{aligned}$$

which is the right hand side of the logical equivalence in the statement of the theorem. In step (a) we applied Sion's minimax theorem (Theorem 2.4.3) which is satisfied because of the assumptions we made. In step (b), we picked Lagrange coefficients, namely $\frac{1}{C_1}$ and $\frac{C_2}{C_1}$, both of which are positive. In particular, C_1^* and C_1 as well as C_2^* and C_2 are related by the Lagrange relaxation equivalence (strong duality). In (c), we multiplied the objective with C_1 throughout, pulled the negative sign in front, and removed the constant terms C_1^* and $C_2 C_2^*$ and used the following observation: $\max_a -g(a) = -\min_a g(a)$; and finally, removed the negative sign in front as this does not affect equivalence. □

The equivalence relationship of Proposition 2.4.1 shows that there is a problem class in which each instance can be viewed either as a RO problem or an estimation problem with an operational cost bias. We can use ideas from RO to make the simultaneous process more general. Before doing so, we will characterize $\mathcal{F}_{good}$ for several specific loss functions.

2.4.2 Creating Uncertainty Sets for RO Using Loss Functions from Machine Learning

Let us for simplicity specialize our loss function to the least squares loss. Let X be an $n \times p$ matrix with each training instance x_i forming the i^{th} row. Also let Y be the n-dimensional vector of all the labels y_i. Then the loss term of (2.1) can be written as:

$$\sum_{i=1}^{n}(y_i - f(x_i))^2 = \sum_{i=1}^{n}(y_i - \beta^T x_i)^2 = \|Y - X\beta\|_2^2.$$

Let β^* be a parameter corresponding to f^* in (2.1). Then the definition of $\mathcal{F}_{good}$ in terms of the least squares loss is:

$$\mathcal{F}_{good} = \{f : f \in \mathcal{F}^R, \|Y - X\beta\|_2^2 \leq \|Y - X\beta^*\|_2^2 + \epsilon\} = \{f : f \in \mathcal{F}^R, \|Y - X\beta\|_2^2 \leq C_1^*\}.$$

Since each $f \in \mathcal{F}_{good}$ corresponds to at least one β, the optimization of (2.1) can be performed with respect to β. In particular, the constraint $\|Y - X\beta\| \leq C_1^*$ is an ellipsoid constraint on β. For the purposes of the robust counterpart in (2.11), we can thus say that the uncertainty is of the ellipsoidal form. In fact, ellipsoidal constraints on uncertain parameters are widely used in robust optimization, especially because the resulting optimization problems often remain tractable.

Box constraints are also a popular way of incorporating uncertainty into robust optimization. For box constraints, the uncertainty over the p-dimensional parameter vector $\beta = [\beta_1, ..., \beta_p]^T$ is written for $i = 1, ..., p$ as $LB_i \leq \beta_i \leq UB_i$, where $\{LB_i\}_i$ and $\{UB_i\}_i$ are real-valued upper and lower bounds that together define the box intervals.

Our main point in this subsection is that one can potentially derive a very wide range of uncertainty sets for robust optimization using different loss functions from machine learning. Box constraints and ellipsoidal constraints are two simple types of

Loss function	Uncertainty set description
least squares	$\|Y - X\beta\|_2^2 \leq \|Y - X\beta^*\|_2^2 + \epsilon$ (ellipsoid)
0-1 loss	$\mathbf{1}_{[f(x_i)\neq y_i]} \leq \mathbf{1}_{[f^*(x_i)\neq y_i]} + \epsilon$
logistic loss	$\sum_{i=1}^n \log(1 + e^{-y_i f(x_i)}) \leq \sum_{i=1}^n \log(1 + e^{-y_i f^*(x_i)}) + \epsilon$
exponential loss	$\sum_{i=1}^n e^{-y_i f(x_i)} \leq \sum_{i=1}^n e^{-y_i f^*(x_i)} + \epsilon$
ramp loss	$\sum_{i=1}^n \min(1, \max(0, 1 - y_i f(x_i))) \leq \sum_{i=1}^n \min(1, \max(0, 1 - y_i f^*(x_i))) + \epsilon$
hinge loss	$\sum_{i=1}^n \max(0, 1 - y_i f(x_i)) \leq \sum_{i=1}^n \max(0, 1 - y_i f^*(x_i)) + \epsilon$

Table 2.1: Table showing a summary of different possible uncertainty set descriptions that are based on ML loss functions.

constraints that could potentially be the set $\mathcal{F}_{good}$, which arise from two different loss functions, as we have shown. The least squares loss leads to ellipsoidal constraints on the uncertainty set, but it is unclear what the structure would be for uncertainty sets arising from the 0-1 loss, ramp loss, hinge loss, logistic loss and exponential loss among others. Further, it is possible to create a loss function for fitting data to a probabilistic model using the method of maximum likelihood; uncertainty sets for maximum likelihood could thus be established. Table 2.4.2 shows several different popular loss functions and the uncertainty sets they might lead to. Many of these new uncertainty sets do not always give tractable mathematical programs, which could explain why they are not commonly considered in the optimization literature.

The sequential process for RO. If we design the uncertainty sets as described above, with respect to a machine learning loss function, the sequential process described in Section 2.2 can be used with robust optimization. This proceeds in three steps:

1. use a learning algorithm on the training data to get f^*,

2. establish an uncertainty set based on the loss function and f^*, for example, ellipsoidal constraints arising from the least squares loss (or one could use any of the new uncertainty sets discussed in the previous paragraph),

3. use specialized optimization techniques to solve for the best policy, with respect to the uncertainty set.

We note that the uncertainty sets created by the 0-1 loss and ramp loss for instance, are non-convex, consequently assumption (**A2**) and Proposition 2.4.1 do not hold for robust optimization problems that use these sets.

2.4.3 The Overlap Between The Simultaneous Process and RO

On the other end of the spectrum from robust optimization, one can think of "optimistic" optimization where we are seeking the best value of the objective in the best possible situation (as oppose to the worst possible situation in RO). For optimistic optimization, more uncertainty is favorable, and we find the best policy for the best possible situation. This could be useful in many real applications where one not only wants to know the worst-case conservative policy but also the best case risk-taking policy. A typical formulation, following (2.11) can be written as:

$$\min_{\pi \in \bigcup_{g \in \mathcal{F}_{good}} \Pi(g;\{\tilde{x}\}_i)} \min_{f \in \mathcal{F}_{good}} \text{OpCost}(\pi, f; \{\tilde{x}_i\}). \qquad (\textit{Optimistic optimization})$$

In optimistic optimization, we view operational cost optimistically ($\min_{f \in \mathcal{F}_{good}}$ OpCost) whereas in the robust optimization counterpart (2.11), we view operational cost conservatively ($\max_{f \in \mathcal{F}_{good}}$ OpCost). The policy π^* is feasible in more situations in RO ($\min_{\pi \in \cap_{g \in \mathcal{F}_{good}} \Pi}$) since it must be feasible with respect to each $g \in \mathcal{F}_{good}$, whereas the OpCost is lower in optimistic optimization ($\min_{\pi \in \cup_{g \in \mathcal{F}_{good}} \Pi}$) since it need only be feasible with respect to at least one of the g's. Optimistic optimization has not been heavily studied, possibly because a (min-min) formulation is relatively easier to solve than its (min-max) robust counterpart, and so is less computationally interesting. Also, one generally plans for the worst case more often than for the best case, particularly when no estimation is involved. In the case where estimation is involved, both optimistic and robust optimization could potentially be useful to a practitioner.

Both optimistic optimization and robust optimization, considered with respect to uncertainty sets $\mathcal{F}_{good}$, have non-trivial overlap with the simultaneous process. In particular, we showed in Proposition 2.4.1 that pessimistic bias on operational

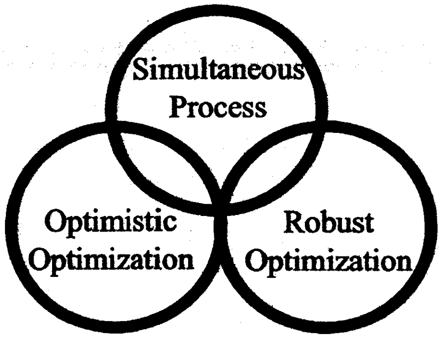


Figure 2-7: Set based description of the proposed framework (top circle) and its relation to robust (right circle) and optimistic (left circle) optimizations. The regions of intersection are where the conditions on the objective OpCost and the feasible set Π are satisfied.

cost is equivalent to robust optimization under specific conditions on OpCost and Π. Using an analogous proof, one can show that optimistic bias on operational cost is equivalent to optimistic optimization under the same set of conditions. Both robust and optimistic optimization and the simultaneous process encompass large classes of problems, some of which overlap. Figure 2-7 represents the overlap between the three classes of problems. There is a class of problems that fall into the simultaneous process, but are not equivalent to robust or optimistic optimization problems. These are problems where we use operational cost to assist with estimation, as in the call center example and ML&TRP discussed in Section 2.3. Typically problems in this class have $\Pi = \Pi(f; \{\tilde{x}_i\}_i)$. This class includes problems where the bias can be either optimistic or pessimistic, and for which F_{good} has a complicated structure, beyond ellipsoidal or box constraints. There are also problems contained in either robust optimization or optimistic optimization alone and do not belong to the simultaneous process. Typically, again, this is when Π depends on f. Note that the housing problem presented in Section 2.3 lies within the intersection of optimistic optimization and the simultaneous process; this can be deduced from (2.7).

In Section 2.5, we will provide statistical guarantees for the simultaneous process. These are very different from the style of probabilistic guarantees in the robust optimization literature. There are some "sample complexity" bounds in the RO literature of the following form: how many observations of uncertain data are required (and applied as simultaneous constraints) to maintain robustness of the solution with high probability? There is an unfortunate overlap in terminology; these are totally different problems to the sample complexity bounds in statistical learning theory. From the learning theory perspective, we ask: how many training instances does it take to come up with a model β that we reasonably know to be good? We will answer that question for a very general class of estimation problems.

2.5 Generalization Bound with New Linear Constraints

In this section, we give statistical learning theoretic results for the simultaneous process that involve counting integer points in convex bodies. Generalization bounds are probabilistic guarantees, that often depend on some measure of the complexity of the hypothesis space. Limiting the complexity of the hypothesis space equates to a better bound. In this section, we consider the complexity of hypothesis spaces that results from an operational cost bias. This enables us to answer in a quantitative manner, question Q3 in the introduction: "Can our intuition about how much it will cost to solve a problem help us produce a better probabilistic model?"

Generalization bounds have been well established for *norm-based* constraints on the hypothesis space, but the emphasis has been more on qualitative dependence (e.g., using big-O notation) and the constants are not emphasized. On the other hand, for a practitioner, every prior belief should reduce the number of examples they need to collect, as these examples may each be expensive to obtain; thus constants within the bounds, and even their approximate values, become important [Bousquet, 2003]. We thus provide bounds on the covering number for new types of hypothesis spaces, emphasizing the role of constants.

To establish the bound, it is sufficient to provide an upper bound on the covering number. There are many existing generic generalization bounds in the literature [e.g., Bartlett and Mendelson, 2002], which combined with our bound, will yield a specific generalization bound for machine learning with operational costs, as we will construct in Theorem 2.5.4.

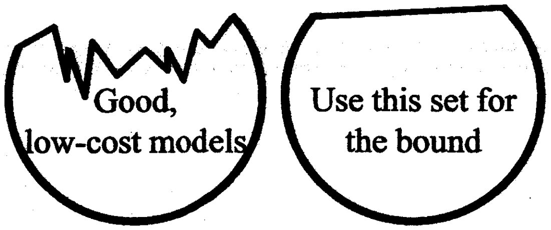


Figure 2-8: Left: hypothesis space for intersection of good models (circular, to represent ℓ_q ball) with low cost models (models below cost threshold, one side of wiggly curve). Right: relaxation to intersection of a half space with an ℓ_q ball.

In Section 2.3, we showed that a bias on the operational cost can sometimes be transformed into linear constraints on model parameter β (see equations (2.5) and (2.8)). There is a broad class of other problems for which this is true, for example, for applications related to those presented in Section 2.3. Because we are able to obtain linear constraints for such a broad class of problems, we will analyze the case of linear constraints here. The hypothesis we consider is thus the intersection of an ℓ_q ball and a halfspace. This is illustrated in Figure 2-8.

The plan for the rest of the section is as follows. We will introduce the quantities on which our main result in this section depends. Then, we will state the main result (Theorem 2.5.1). Following that, we will build up to a generalization bound (Theorem 2.5.4) that incorporates Theorem 2.5.1. After that will be the proof of Theorem 2.5.1.

Definition 2. *[Covering Number, Kolmogorov and Tikhomirov, 1959]* Let $A \subseteq \Gamma$ be an arbitrary set and (Γ, ρ) a (pseudo-)metric space. Let $|\cdot|$ denote set size.

- For any $\epsilon > 0$, an **ϵ-cover** for A is a finite set $U \subseteq \Gamma$ (not necessarily $\subseteq A$) s.t. $\forall a \in A, \exists u \in U$ with $d_\rho(a, u) \leq \epsilon$.
- The **covering number** of A is $N(\epsilon, A, \rho) := \inf_U |U|$ where U is an ϵ-cover for A.

We are given the set of n instances $S := \{x_i\}_{i=1}^n$ with each $x_i \in \mathcal{X} \subseteq \mathbb{R}^p$ where $\mathcal{X} = \{x : \|x\|_r \leq X_b\}$, $2 \leq r \leq \infty$ and X_b is a known constant. Let $\mu_{\mathcal{X}}$ be a probability measure on $\mathcal{X}$. Let x_i be arranged as rows of a matrix X. We can represent the columns of $X = [x_1 \; \dots \; x_n]^T$ with $h_j \in \mathbb{R}^n, j = 1, ..., p$, so X can also be written as $[h_1 \cdots h_p]$. Define function class $\mathcal{F}$ as the set of linear functionals whose coefficients lie in an ℓ_q ball and with a set of linear constraints:

$$\mathcal{F} := \{f : f(x) = \beta^T x, \beta \in \mathcal{B}\} \text{ where}$$
$$\mathcal{B} := \left\{\beta \in \mathbb{R}^p : \|\beta\|_q \leq B_b, \sum_{j=1}^{p} c_{j\nu}\beta_j + \delta_\nu \leq 1, \delta_\nu > 0, \nu = 1, ..., V\right\},$$

where $1/r + 1/q = 1$ and $\{c_{j\nu}\}_{j,\nu}$, $\{\delta_\nu\}_\nu$ and B_b are known constants. The linear constraints given by the $c_{j\nu}$'s force the hypothesis space $\mathcal{F}$ to be smaller, which will help with generalization - this will be shown formally by our main result in this section. Let $\mathcal{F}_{|S}$ be defined as the restriction of $\mathcal{F}$ with respect to S.

Let $\{\tilde{c}_{j\nu}\}_{j,\nu}$ be proportional to $\{c_{j\nu}\}_{j,\nu}$:

$$\tilde{c}_{j\nu} \; := \; \frac{c_{j\nu} n^{1/r} X_b B_b}{\|h_j\|_r} \quad \forall j = 1, ..., p \text{ and } \nu = 1, ..., V.$$

Let K be a positive number. Further, let the sets P^K parameterized by K and P_c^K parameterized by K and $\{\tilde{c}_{j\nu}\}_{j,\nu}$ be defined as

$$P^K := \left\{(k_1, ..., k_p) \in \mathbb{Z}^p : \sum_{j=1}^{p} |k_j| \leq K\right\}.$$
$$P_c^K := \left\{(k_1, ..., k_p) \in P^K : \sum_{j=1}^{p} \tilde{c}_{j\nu} k_j \leq K \; \forall \nu = 1, ..., V\right\}. \tag{2.15}$$

Let $|P^K|$ and $|P_c^K|$ be the sizes of the sets P^K and P_c^K respectively. The subscript c in P_c^K denotes that this polyhedron is a constrained version of P^K. As the linear constraints given by the $c_{j\nu}$'s force the hypothesis space to be smaller, they force $|P_c^K|$ to be smaller. Define X_{sL} to be equal to X times a diagonal matrix whose j^{th} diagonal element is $\frac{n^{1/r} X_b B_b}{\|h_j\|_r}$. Define $\lambda_{\min}(X_{sL}{}^T X_{sL})$ to be the smallest eigenvalue of the matrix $X_{sL}{}^T X_{sL}$, which will thus be non-negative. Using these definitions, we state our main result of this section.

Theorem 2.5.1. (Main result, covering number bound)

$$N(\sqrt{n}\epsilon, \mathcal{F}_{|S}, \|\cdot\|_2) \leq \begin{cases} \min\{|P^{K_0}|, |P_c^K|\} & \text{if } \epsilon < X_b B_b \\ 1 & \text{otherwise} \end{cases}, \tag{2.16}$$

where

$$K_0 = \left\lceil \frac{X_b^2 B_b^2}{\epsilon^2} \right\rceil$$

and

$$K = \max\left\{ K_0, \left\lceil \frac{n X_b^2 B_b^2}{\lambda_{\min}({X_{sL}}^T X_{sL}) \left[\min_{\nu=1,\dots,V} \frac{\delta_\nu}{\sum_{j=1}^p |\tilde{c}_{j\nu}|} \right]^2} \right\rceil \right\}.$$

The theorem gives a bound on the ℓ_2 covering number for the specially constrained class $\mathcal{F}_{|S}$. The bound improves as the constraints given by $c_{j\nu}$ on the operational cost become tighter. In other words, as the $c_{j\nu}$ impose more restrictions on the hypothesis space, $|P_c^K|$ decreases, and the covering number bound becomes smaller. This bound can be plugged directly into an established generalization bound that incorporates covering numbers, and this is done in what follows to obtain Theorem 2.5.4.

Note that $\min\{|P^{K_0}|, |P_c^K|\}$ can be tighter than $|P_c^K|$ when ϵ is large. When ϵ is larger than $X_b B_b$, we only need one closed ball of radius $\sqrt{n}\epsilon$ to cover $\mathcal{F}_{|S}$, so $N(\sqrt{n}\epsilon, \mathcal{F}_{|S}, \|\cdot\|_2) = 1$. In that case, the covering number in Theorem 2.5.1 is appropriately bounded by 1. If ϵ is large, but not larger than $X_b B_b$, then $|P_c^K|$ can be smaller than $|P^{K_0}|$. $|P^{K_0}|$ is the size of the polytope without the operational cost constraints. $|P_c^K|$ is the size of a potentially bigger polytope, but with additional constraints.

For this problem we generally assume that $n > p$; that is the number of examples is greater than the dimensionality p. In such a case, $\lambda_{\min}({X_{sL}}^T X_{sL})$ can be shown to be bounded away from zero for a wide variety of distributions $\mu_{\mathcal{X}}$ (e.g., sub-gaussian zero-mean). When $\lambda_{\min}({X_{sL}}^T X_{sL}) = 0$, the covering number bound becomes vacuous.

Let us introduce some notation in order to state the generalization bound results.

Given any function $f \in \mathcal{F}$, we would like to minimize the expected future loss (also known as the expected risk), defined as:

$$R^{\text{true}}(l \circ f) := \mathbb{E}_{(x,y)\sim\mu_{\mathcal{X}\times\mathcal{Y}}}\Big[l(f(x),y)\Big] = \int l(f(x),y)\partial\mu_{\mathcal{X}\times\mathcal{Y}}(x,y),$$

where $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is the (fixed) loss function we had previously defined in Section 2.2. The loss on the training sample (also known as the empirical risk) is:

$$R^{\text{emp}}(l \circ f, \{(x_i, y_i)\}_1^n) := \frac{1}{n}\sum_{i=1}^{n} l(f(x_i), y_i).$$

We would like to know that $R^{\text{true}}(l \circ f)$ is not too much more than $R^{\text{emp}}(l \circ f, \{(x_i, y_i)\}_1^n)$, no matter which f we choose from $\mathcal{F}$. A typical form of generalization bound that holds with high probability for every function in $\mathcal{F}$ is

$$R^{\text{true}}(l \circ f) \leq R^{\text{emp}}(l \circ f, \{(x_i, y_i)\}_1^n) + \text{Bound}(\text{complexity}(\mathcal{F}),n), \tag{2.17}$$

where the complexity term takes into account the constraints on $\mathcal{F}$, both the linear constraints, and the ℓ_q-ball constraint. Theorem 2.5.1 gives an upper bound on the term Bound(complexity($\mathcal{F}$),n) in (2.17) above. In order to show this explicitly, we will give the definition of Rademacher complexity, restate how it appears in the relation between expected future loss and loss on training examples, and state an upper-bound for it in terms of the covering number.

Definition 3. (Rademacher Complexity) The empirical Rademacher complexity of $\mathcal{F}_{|S}$ is[5]

$$\hat{\mathcal{R}}(\mathcal{F}_{|S}) = \mathbb{E}_\sigma\left[\sup_{f\in\mathcal{F}} \frac{2}{n}\sum_{i=1}^{n} \sigma_i f(x_i)\right] \tag{2.18}$$

where $\{\sigma_i\}$ are Rademacher random variables ($\sigma_i = 1$ with prob. $1/2$ and -1 with prob. $1/2$). The Rademacher complexity is its expectation: $\mathcal{R}_n(\mathcal{F}) = \mathbb{E}_{S\sim(\mu_\mathcal{X})^n}[\hat{\mathcal{R}}(\mathcal{F}_{|S})]$.

The empirical Rademacher complexity $\hat{\mathcal{R}}(\mathcal{F}_{|S})$ can be computed given S and $\mathcal{F}$,

[5] The factor 2 in the defining equation (2.18) is not very important. Some authors omit this factor and include it explicitly as a pre-factor in, for example, Theorem 2.5.2.

and by concentration, will be close to the Rademacher complexity. The following result relates the true risk to the empirical risk and empirical Rademacher complexity for any function class $\mathcal{H}$ [see Bartlett and Mendelson, 2002, and references therein]. Let the quantities $\mathcal{H}_{|S}, R^{true}(l \circ h)$ and $R^{\text{emp}}(l \circ h, \{x_i, y_i\}_1^n)$ be analogous to those we had defined for our specific class $\mathcal{F}$.

Theorem 2.5.2. (Rademacher Generalization Bound) For all $\delta > 0$, with probability at least $1 - \delta, \forall h \in \mathcal{H}$,

$$R^{\text{true}}(l \circ h) \leq R^{\text{emp}}(l \circ h, \{x_i, y_i\}_1^n) + \mathcal{L} \cdot \hat{\mathcal{R}}(\mathcal{H}_{|S}) + \frac{3}{\sqrt{2}}\sqrt{\frac{\log \frac{1}{\delta}}{n}}, \tag{2.19}$$

where $\mathcal{L}$ is the Lipschitz constant of the loss function.

Note that (2.19) is an explicit form of (2.17). We will now relate $\hat{\mathcal{R}}(\mathcal{F}_{|S})$ to covering numbers thus justifying the importance of statement (2.16) in Theorem 2.5.1. In particular the following infinite chaining argument also known as Dudley's integral [see Talagrand, 2005] relates $\hat{\mathcal{R}}(\mathcal{F}_{|S})$ to the covering number of the set $\mathcal{F}_{|S}$.

Theorem 2.5.3. (Relating Rademacher Complexity to Covering Numbers) We are given that $\forall x \in \mathcal{X}$, we have $f(x) \in [-X_b B_b, X_b B_b]$. Then,

$$\frac{1}{X_b B_b}\hat{\mathcal{R}}(\mathcal{F}_{|S}) \leq 12 \int_0^\infty \sqrt{\frac{2 \log N(\alpha, \mathcal{F}, L_2(\mu_{\mathcal{X}}^n))}{n}} d\alpha = 12 \int_0^\infty \sqrt{\frac{2 \log N(\sqrt{n}\alpha, \mathcal{F}_{|S}, \| \cdot \|_2)}{n}} d\alpha.$$

Our main result in Theorem 2.5.1 can be used in conjunction with Theorems 2.5.2 and 2.5.3, to directly see how the true error relates to the empirical error and the constraints on the restricted function class $\mathcal{F}$ (the ℓ_q-norm bound on β and linear constraint on β from the operational cost bias). Explicitly, that bound is here.

Theorem 2.5.4. (Generalization Bound for ML with Operational Costs) For all $\delta > 0$, with probability at least $1 - \delta, \forall f \in \mathcal{F}$,

$$R^{\text{true}}(l \circ f) \leq R^{\text{emp}}(l \circ f, \{x_i, y_i\}_1^n) + 12 \mathcal{L} X_b B_b \int_0^\infty \sqrt{\frac{2 \log N(\sqrt{n}\epsilon, \mathcal{F}_{|S}, \| \cdot \|_2)}{n}} d\epsilon + \frac{3}{\sqrt{2}}\sqrt{\frac{\log \frac{1}{\delta}}{n}},$$

where

$$N(\sqrt{n}\epsilon, \mathcal{F}_{|S}, \|\cdot\|_2) \leq \begin{cases} \min\{|P^{K_0}|, |P_c^K|\} & \text{if } \epsilon < X_b B_b \\ 1 & \text{otherwise} \end{cases},$$

$$K_0 = \left\lceil \frac{X_b^2 B_b^2}{\epsilon^2} \right\rceil,$$

and

$$K = \max\left\{ K_0, \left\lceil \frac{n X_b^2 B_b^2}{\lambda_{\min}({X_{sL}}^T X_{sL}) \left[\min_{\nu=1,\ldots,V} \frac{\delta_\nu}{\sum_{j=1}^p |\tilde{c}_{j\nu}|} \right]^2} \right\rceil \right\}$$

are functions of ϵ.

This bound implies that prior knowledge about the operational cost can be important for generalization. As our prior knowledge on the cost becomes stronger, the size of the hypothesis space becomes more restrictive, as seen through the constraints given by the $c_{j\nu}$. When this happens, the $|P_c^K|$ terms become smaller, and the whole bound becomes smaller. Note that the integral over ϵ is taken from $\epsilon = 0$ to $\epsilon = \infty$. When ϵ is larger than $X_b B_b$, as noted earlier, $N(\sqrt{n}\epsilon, \mathcal{F}_{|S}, \|\cdot\|_2) = 1$ and thus $\log N(\sqrt{n}\epsilon, \mathcal{F}_{|S}, \|\cdot\|_2) = 0$.

Before we move onto building the necessary tools to prove Theorem 2.5.1, we compare our result with the bound in our work on the ML&TRP [Tulabandhula et al., 2011]. In that work, we considered a linear function class with a constraint on the ℓ_2-norm and one additional linear inequality constraint on β. We then used a sample independent volumetric cap argument to get a covering number bound. Theorem 2.5.1 is in some ways an improvement of the other result: (1) we can now have multiple linear constraints on β; (2) our new result involves a sample-specific bounding technique for covering numbers, which is generally tighter; (3) our result applies to ℓ_q balls for $q \in [1, 2]$ whereas the previous analysis holds only for $q = 2$. The volumetric argument in [Tulabandhula et al., 2011] provided a scaling of the covering number. Specifically, the operational cost term for the ML&TRP allowed us to reduce the covering number term in the bound from $\sqrt{\log N(\cdot,\cdot,\cdot)}$ to $\sqrt{\log(\alpha N(\cdot,\cdot,\|\cdot\|_2))}$, or equivalently $\sqrt{\log N(\cdot,\cdot,\|\cdot\|_2)+\log\alpha}$, where α is a function of the operational cost constraint. If α obeys $\alpha \ll 1$, then there is a noticeable effect on the generalization bound, compared to almost no effect when $\alpha \approx 1$. In the present work, the bound does not scale the covering number like this, instead it is a very different approach giving a more direct bound.

2.5.1 Proof of Theorem 2.5.1

We make use of Maurey's Lemma [Barron, 1993] in our proof [in the same spirit as Zhang, 2002]. The main ideas of Maurey's Lemma are used in many machine learning papers in various contexts [*e.g.*, Koltchinskii and Panchenko, 2005, Schapire et al., 1998, Rudin and Schapire, 2009]. Our proof of Theorem 2.5.1 adapts Maurey's Lemma to handle polyhedrons, and allows us to apply counting techniques to bound the covering number.

Recall that $X = [x_1 \dots x_n]^T$ was also defined column-wise as $[h_1 \ \dots \ h_p]$. We introduce two scaled sets $\{\tilde{h}_j\}_j$ and $\{\tilde{\beta}_j\}_j$ corresponding to $\{h_j\}_j$ and $\{\beta_j\}_j$ as follows:

$$\tilde{h}_j := \frac{n^{1/r}X_bB_b}{\|h_j\|_r}h_j \text{ for } j = 1,...,p; \text{ and}$$
$$\tilde{\beta}_j := \frac{\|h_j\|_r}{n^{1/r}X_bB_b}\beta_j \text{ for } j = 1,...,p.$$

These scaled sets will be convenient in places where we do not want to carry the scaling terms separately.

Any vector y that is equal to $X\beta$ can thus be written in three different ways:

$$y = \sum_{j=1}^{p} \beta_j h_j, \text{ or}$$
$$y = \sum_{j=1}^{p} \tilde{\beta}_j \tilde{h}_j, \text{ or}$$
$$y = \sum_{j=1}^{p} |\tilde{\beta}_j| \text{sign}(\tilde{\beta}_j)\tilde{h}_j.$$

Our first lemma is a restatement of Maurey's lemma [revised version of Lemma

1 in Zhang, 2002]. We provide a proof based on the law of large numbers [Barron, 1993] though other proof techniques also exist [see Jones, 1992, for a proof based on iterative approximation].

The lemma states that every point y in the convex hull of $\{h_j\}_j$ is close to one of the points y_K in a particular finite set.

Lemma 2.5.5. Let $\max_{j=1,\dots,p} \|\tilde{h}_j\|$ be less than or equal to some constant b. If y belongs to the convex hull of set $\{\tilde{h}_j\}_j$, then for every positive integer $K \geq 1$, there exists y_K in the convex hull of K points of set $\{\tilde{h}_j\}_j$ such that $\|y - y_K\|^2 \leq \frac{b^2}{K}$.

Proof. Let y be written in the form:

$$y = \sum_{i=1}^{p} \bar{\gamma}_j \tilde{h}_j,$$

where for each $j = 1, \dots, p$, $\bar{\gamma}_j \geq 0$ and $\sum_{j=1}^{p} \bar{\gamma}_j \leq 1$. Let $\bar{\gamma}_{p+1} := 1 - \sum_{j=1}^{p} \bar{\gamma}_j$.

Consider a discrete distribution $\mathcal{D}$ formed by the coefficient vector $(\bar{\gamma}_1, .., \bar{\gamma}_p, \bar{\gamma}_{p+1})$. Associate a random variable $\tilde{\mathsf{h}}$ with support set $\{\tilde{h}_1, \dots, \tilde{h}_p, \mathbf{0}\}$. That is, $\Pr(\tilde{\mathsf{h}} = \tilde{h}_j) = \bar{\gamma}_j$, $j = 1, \dots, p$ and $\Pr(\tilde{\mathsf{h}} = \mathbf{0}) = \bar{\gamma}_{p+1}$.

Draw K observations $\{\tilde{h}^1, \dots, \tilde{h}^K\}$ uniformly and independently from $\mathcal{D}$ and form the sample average $\mathsf{y}_K := \frac{1}{K}\sum_{s=1}^{K} \tilde{h}^s$. Here, we are using the superscript index to denote the observation number. The mean of this random variable y_K is:

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}}[\mathsf{y}_K] &= \frac{1}{K}\sum_{s=1}^{K} \mathbb{E}_{\mathcal{D}}[\tilde{h}^s] \text{ where} \\
\mathbb{E}_{\mathcal{D}}[\tilde{h}^s] &= \sum_{j=1}^{p+1} \Pr(\tilde{\mathsf{h}} = \tilde{h}_j)\tilde{h}_j = \sum_{j=1}^{p} \bar{\gamma}_j \tilde{h}_j = y
\end{aligned}$$

hence $\mathbb{E}_{\mathcal{D}}[\mathsf{y}_K] = y$.

The expected distance between y_K and y is:

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}}[\|\mathsf{y}_K - y\|^2] &= \mathbb{E}_{\mathcal{D}}[\|\mathsf{y}_K - \mathbb{E}_{\mathcal{D}}[\mathsf{y}_K]\|^2] = \mathbb{E}\left[\sum_{i=1}^{n} (\mathsf{y}_K - \mathbb{E}_{\mathcal{D}}[\mathsf{y}_K])_i^2\right] \\
&\overset{(\dagger)}{=} \sum_{i=1}^{n} \mathrm{Var}((\mathsf{y}_K)_i) \overset{(*)}{=} \sum_{i=1}^{n} \frac{1}{K}\mathrm{Var}((\tilde{\mathsf{h}})_i)
\end{aligned}$$

$$\begin{aligned} &\overset{(\ddagger)}{=} \frac{1}{K}\sum_{i=1}^{n}\left(\mathbb{E}_{\mathcal{D}}[(\tilde{\mathsf{h}})_i^2] - \mathbb{E}_{\mathcal{D}}[(\tilde{\mathsf{h}})_i]^2\right) \overset{(\circ)}{=} \frac{1}{K}\left(\mathbb{E}_{\mathcal{D}}[\|\tilde{\mathsf{h}}\|^2] - \|\mathbb{E}_{\mathcal{D}}[\tilde{\mathsf{h}}]\|^2\right) \\ &\leq \frac{1}{K}\mathbb{E}_{\mathcal{D}}[\|\tilde{\mathsf{h}}\|^2] \leq \frac{b^2}{K} \end{aligned} \tag{2.20}$$

where we have used i to be the index for the i^{th} coordinate of the n dimensional vectors. (†) follows from the definition of variance coordinate-wise. ($*$) follows because each component of y_K is a sample average. (‡) also follows from the definition of variance. At step ($\circ$), we rewrite the previous summations involving squares into ones that use the Hilbert norm. Our assumption on $\max_{j=1,\dots,p}\|\tilde{h}_j\|$ tells us that $\mathbb{E}_{\mathcal{D}}[\|\tilde{\mathsf{h}}\|^2] \leq b^2$ leading to (2.20). Since the squared Hilbert norm of the sample mean is bounded in this way, there exists a y_K that satisfies the inequality, so that

$$\|y_K - y\|^2 \leq \frac{b^2}{K}.$$

□

The following corollary states explicitly that an approximation to y exists that is a linear combination with coefficients chosen from a particular discrete set.

Corollary 2.5.6. For any y and K as considered above, we can find non-negative integers $m_1, ..., m_p$ such that $\sum_{j=1}^{p} m_j \leq K$ and $\|y - \sum_{j=1}^{p}\frac{m_j}{K}\tilde{h}_j\|^2 \leq \frac{b^2}{K}$.

This follows immediately from the proof of Lemma 2.5.5, choosing m_j to be the coefficients of the $\tilde{h}_j$'s such that $y_K = \sum_j \frac{m_j}{K}\tilde{h}_j$.

The above corollary means that counting the number of p-tuple non-negative integers $m_1, ..., m_p$ gives us a covering of the set that y belongs to. In the case of Lemma 2.5.5, this set is the convex hull of $\{\tilde{h}_j\}_j$.

Before we can go further, we need to generalize the argument from the positive orthant of the ℓ_1 ball to handle any coefficients that are in the whole unit-length ℓ_1-ball. This is what the following lemma accomplishes.

Lemma 2.5.7. Let $\max_{j=1,\dots,p}\|\tilde{h}_j\|$ be less than or equal to some constant b. For any $y = \sum_{j=1}^{p}\tilde{\beta}_j\tilde{h}_j$ such that $\|\tilde{\beta}\|_1 \leq 1$, given a positive integer K, we can find a y_K such that

$$\|y - y_K\|_2^2 \leq \frac{b^2}{K}$$

where $y_K = \sum_{j=1}^{p} \frac{k_j}{K}\tilde{h}_j$ is a combination of $\{\tilde{h}_j\}$ with integers $k_1, ..., k_p$ such that $\sum_{j=1}^{p} |k_j| \leq K$.

Proof. Lemma 2.5.5 cannot be applied directly since the $\{\tilde{\beta}_j\}_j$ can be negative. We rewrite y or equivalently $\sum_{j=1}^{p} \tilde{\beta}_j\tilde{h}_j$ as

$$y = \sum_{j=1}^{p} |\tilde{\beta}_j|\text{sign}(\tilde{\beta}_j)\tilde{h}_j.$$

Thus y lies in the convex combination of $\{\text{sign}(\tilde{\beta}_j)\tilde{h}_j\}_j$. Note that this step makes the convex hull depend on the y or $\{\tilde{\beta}_j\}_j$ we start with. Nonetheless, we know by substituting $\{\text{sign}(\tilde{\beta}_j)\tilde{h}_j\}_j$ for $\{\tilde{h}_j\}_j$ in the statement of Lemma 2.5.5 and Corollary 2.5.6 that

1. we can find y_K, or equivalently

2. we can find non-negative integers $m_1, ..., m_p$ with $\sum_{j=1}^{p} m_j \leq K$,

such that $\|y - y_K\|_2^2 \leq \frac{b^2}{K}$ where $y_K = \sum_{j=1}^{p} \frac{m_j}{K}\text{sign}(\tilde{\beta}_j)\tilde{h}_j$ holds. This implies there exist integers $k_1, ..., k_p$ such that $y_K = \sum_{j=1}^{p} \frac{k_j}{K}\tilde{h}_j$ where $\sum_{j=1}^{p} |k_j| \leq K$. We simply let $k_j = m_j\text{sign}(\tilde{\beta}_j)$. Thus, we absorbed the signs of the $\tilde{\beta}_j$'s, and the coefficients no longer need to be nonnegative.

In other words, we have shown that if a particular y_K is in the convex hull of points $\{\text{sign}(\tilde{\beta}_j)\tilde{h}_j\}_j$, then the same y_K is a linear combination of $\{\tilde{h}_j\}_j$ where the coefficients of the combination $k_1/K, ..., k_p/K$ obey $\sum_{j=1}^{p} |k_j| \leq K$. This concludes the proof. □

We now want to answer the question of whether the $k_1/K, ..., k_p/K$ can obey (related) linear constraints if the original $\{\tilde{\beta}_j\}_j$ did so. These constraints on the $\{\tilde{\beta}_j\}_j$'s are the ones coming from constraints on the operational cost. In other words, we want to know that our (discretized) approximation of y also obeys a constraint coming from the operational cost.

Let $\{\tilde{\beta}_j\}_j$ satisfy the linear constraints within the definition of $\mathcal{B}$, in addition to satisfying $\|\tilde{\beta}\|_1 \leq 1$:

$$\sum_{j=1}^{p} \tilde{c}_{j\nu}\tilde{\beta}_j + \delta_\nu \leq 1, \text{ for fixed } \delta_\nu > 0, \nu = 1, ..., V.$$

We now want that for large enough K, the p-tuple $k_1/K, ..., k_p/K$ also meets certain related linear constraints.

We will make use of the matrix X_{sL}, defined before Theorem 2.5.1. It has the elements of the scaled set $\{\tilde{h}_j\}_j$ as its columns: $X_{sL} := [\tilde{h}_1 \ \ldots \ \tilde{h}_p]$.

Lemma 2.5.8. Take any $y = \sum_{j=1}^{p} \tilde{\beta}_j \tilde{h}_j$, and any $y_K = \sum_{j=1}^{p} \frac{k_j}{K}\tilde{h}_j$, with:

$$\sum_{j=1}^{p} \tilde{c}_{j\nu}\tilde{\beta}_j + \delta_\nu \leq 1, \text{ for fixed } \delta_\nu > 0, \nu = 1, ..., V \text{ where } \|\tilde{\beta}\|_1 \leq 1$$

and $\|y - y_K\|_2^2 \leq b^2/K$. Whenever

$$K \geq \frac{b^2}{\left[\min_{\nu=1,...,V} \frac{\delta_\nu}{\sum_{j=1}^{p} |\tilde{c}_{j\nu}|}\right]^2 \lambda_{\min}(X_{sL}{}^T X_{sL})},$$

then the following linear constraints on $k_1/K, ..., k_p/K$ hold:

$$\sum_{j=1}^{p} \tilde{c}_{j\nu}\frac{k_j}{K} \leq 1, \ \nu = 1, ..., V.$$

This lemma states that as long as the discretization is fine enough, our approximation y_K obeys similar operational cost constraints to y.

Proof. Let $\kappa := [k_1/K \ \ldots \ k_p/K]^T$. Using the definition of X_{sL},

$$\frac{b^2}{K} \geq \|y - y_K\|_2^2 = \|X_{sL}\tilde{\beta} - X_{sL}\kappa\|_2^2 = \|X_{sL}(\tilde{\beta} - \kappa)\|_2^2$$

$$= (\tilde{\beta} - \kappa)^T {X_{sL}}^T X_{sL}(\tilde{\beta} - \kappa) \overset{(*)}{\geq} \lambda_{\min}({X_{sL}}^T X_{sL})\|\tilde{\beta} - \kappa\|_2^2. \tag{2.21}$$

In $(*)$, we used the fact that for a positive (semi-)definite matrix M and for every non-zero vector z, $z^T M z \geq \lambda_{\min}(M) z^T I z$. (If $\tilde{\beta} = \kappa$, we are done since κ will obey the constraints $\tilde{\beta}$ obeys.) Also, for any z, in each coordinate j, $|z_j| \leq \max_{j=1,\ldots,p} |z_j| = \|z\|_\infty \leq \|z\|_2$. Combining this with (2.21), we have:

$$\left|\tilde{\beta}_j - \frac{k_j}{K}\right| \leq \|\tilde{\beta} - \kappa\|_2 \leq \frac{b}{\sqrt{K\lambda_{\min}({X_{sL}}^T X_{sL})}}.$$

This implies that κ itself component-wise satisfies

$$\tilde{\beta}_j - A \leq \frac{k_j}{K} \leq \tilde{\beta}_j + A \text{ where } A := \frac{b}{\sqrt{K\lambda_{\min}({X_{sL}}^T X_{sL})}}.$$

So far we know that for all $\nu = 1, \ldots, V$, $\sum_{j=1}^p \tilde{c}_{j\nu}\tilde{\beta}_j + \delta_\nu \leq 1$, with $\delta_\nu > 0$, and each coordinate k_j/K within κ varies from $\tilde{\beta}_j$ by at most an amount A. We would like to establish that the linear constraints $\sum_{j=1}^p \tilde{c}_{j\nu}\frac{k_j}{K} \leq 1$, $\nu = 1, \ldots, V$; always hold for such a κ. For each constraint ν, substituting the extremal values of k_j according to the sign of $\tilde{c}_{j\nu}$, we get the following upper bound:

$$\sum_{j=1}^p \tilde{c}_{j\nu}\frac{k_j}{K} \leq \sum_{\tilde{c}_{j\nu}>0} \tilde{c}_{j\nu}(\tilde{\beta}_j + A) + \sum_{\tilde{c}_{j\nu}<0} \tilde{c}_{j\nu}(\tilde{\beta}_j - A) = \sum_{j=1}^p \tilde{c}_{j\nu}\tilde{\beta}_j + A\sum_{j=1}^p |\tilde{c}_{j\nu}|.$$

This sum $\sum_{j=1}^p \tilde{c}_{j\nu}\tilde{\beta}_j + A\sum_{j=1}^p |\tilde{c}_{j\nu}|$ is less than or equal to 1 iff $A\sum_{j=1}^p |\tilde{c}_{j\nu}| \leq \delta_\nu$. Thus we would like $A \leq \frac{\delta_\nu}{\sum_{j=1}^p |\tilde{c}_{j\nu}|}$ for all $\nu = 1, \ldots, V$. That is,

$$\frac{b}{\sqrt{K\lambda_{\min}({X_{sL}}^T X_{sL})}} = A \leq \min_{\nu=1,\ldots,V} \frac{\delta_\nu}{\sum_{j=1}^p |\tilde{c}_{j\nu}|}$$

$$\Leftrightarrow \quad K \geq \frac{b^2}{\left[\min_{\nu=1,\ldots,V} \frac{\delta_\nu}{\sum_{j=1}^p |\tilde{c}_{j\nu}|}\right]^2 \lambda_{\min}({X_{sL}}^T X_{sL})}.$$

□

We now proceed with the proof of our main result of this section. The result involves covering numbers, where the cover for the set will be the vectors with discretized coefficients that we have been working with in the lemmas above.

Proof. (of Theorem 2.5.1)

Recall that

- the matrix X is defined as $[h_1 \ldots h_p]$;
- the scaled versions of vector $\{h_j\}_j$ are $\tilde{h}_j = \frac{n^{1/r} X_b B_b}{\|h_j\|_r} h_j$ for $j = 1, \ldots, p$;
- the scaled versions of coefficients $\{\beta_j\}_j$ are $\tilde{\beta}_j = \frac{\|h_j\|_r}{n^{1/r} X_b B_b} \beta_j$ for $j = 1, \ldots, p$; and
- any vector $y = X\beta = \sum_{j=1}^{p} \beta_j h_j$ can be rewritten as $\sum_{j=1}^{p} \tilde{\beta}_j \tilde{h}_j$.

We will prove three technical facts leading up to the result.

Fact 1. If $\|\beta\|_q \leq B_b$, then $\|\tilde{\beta}\|_1 \leq 1$.

Because $1/r + 1/q = 1$, by Hölder's inequality we have:

$$\begin{aligned} \sum_{j=1}^{p} |\tilde{\beta}_j| &= \frac{1}{n^{1/r} B_b X_b} \sum_{j=1}^{p} \|h_j\|_r |\beta_j| \\ &\leq \frac{1}{n^{1/r} B_b X_b} \left(\sum_{j=1}^{p} \|h_j\|_r^r \right)^{1/r} \left(\sum_{j=1}^{p} |\beta^j|^q \right)^{1/q}. \end{aligned} \tag{2.22}$$

To bound the above notice that in our notation, $(h_j)_i = (x_i)_j$. That is, the i^{th} component of feature vector h_j, i.e., $(h_j)_i$ is also the j^{th} component of example x_i. Thus,

$$\begin{aligned} \left(\sum_{j=1}^{p} \|h_j\|_r^r \right)^{1/r} &= \left(\sum_{j=1}^{p} \sum_{i=1}^{n} ((h_j)_i)^r \right)^{1/r} = \left(\sum_{i=1}^{n} \sum_{j=1}^{p} ((h_j)_i)^r \right)^{1/r} \\ &= \left(\sum_{i=1}^{n} \|x_i\|_r^r \right)^{1/r} \leq (n X_b^r)^{1/r} = n^{1/r} X_b. \end{aligned}$$

Plugging this into (2.22), and using the fact that $\|\beta\|_q \leq B_b$, we have

$$\sum_{j=1}^{p} |\tilde{\beta}_j| \leq \frac{1}{n^{1/r} B_b X_b} n^{1/r} X_b B_b = 1,$$

that is, $\|\tilde{\beta}\|_1 \leq 1$.

Fact 2. Corresponding to the set of linear constraints on β:

$$\sum_{j=1}^{p} c_{j\nu}\beta_j + \delta_\nu \leq 1, \delta_\nu > 0, \nu = 1, ..., V,$$

there is a set of linear constraints on $\tilde{\beta}_j$, namely $\sum_{j=1}^{p} \tilde{c}_{j\nu}\tilde{\beta}_j + \delta_\nu \leq 1, \nu = 1, ..., V$.

Recall that $\beta \in \mathcal{B}$ also means that $\sum_{j=1}^{p} c_{j\nu}\beta_j + \delta_\nu \leq 1$ for some $\delta_\nu > 0$ for all $\nu = 1, ..., V$. Thus, for all $\nu = 1, ..., V$:

$$\begin{aligned} & \sum_{j=1}^{p} c_{j\nu}\beta_j + \delta_\nu \leq 1 \\ \Leftrightarrow \quad & \sum_{j=1}^{p} c_{j\nu} \left(\frac{n^{1/r} X_b B_b}{\|h_j\|_r} \frac{\|h_j\|_r}{n^{1/r} X_b B_b} \right) \beta_j + \delta_\nu \leq 1 \\ \Leftrightarrow \quad & \sum_{j=1}^{p} \tilde{c}_{j\nu}\tilde{\beta}_j + \delta_\nu \leq 1 \end{aligned}$$

which is the set of corresponding linear constraints on $\{\tilde{\beta}_j\}_j$ we want.

Fact 3. $\forall j = 1, ..., p,\ \|\tilde{h}_j\|_2 \leq n^{1/2} X_b B_b$.

Jensen's inequality implies that for any vector z in $\mathbb{R}^n$, and for any $r \geq 2$, it is true that $\frac{1}{n^{1/2}}\|z\|_2 \leq \frac{1}{n^{1/r}}\|z\|_r$. Using this for our particular vector $\tilde{h}_j$ and our given r, we get

$$\|\tilde{h}_j\|_2 \leq \|\tilde{h}_j\|_r n^{1/2} \frac{1}{n^{1/r}}.$$

But we know

$$\|\tilde{h}_j\|_r = \left\| \frac{n^{1/r} X_b B_b}{\|h_j\|_r} h_j \right\|_r = \frac{n^{1/r} X_b B_b}{\|h_j\|_r} \|h_j\|_r = n^{1/r} X_b B_b.$$

Thus, we have $\|\tilde{h}_j\|_2 \leq n^{1/2} X_b B_b$ for each j, and thus, $\max_{j=1,...,p} \|\tilde{h}_j\|_2 \leq n^{1/2} X_b B_b$.

With those three facts established, we can proceed with the proof of Theorem 2.5.1. Facts 1 and 2 show that the requirements on $\tilde{\beta}$ for Lemma 2.5.7 and Lemma 2.5.8 are satisfied. Fact 3 shows that the requirement on $\{\tilde{h}_j\}_j$ for Lemma 2.5.7 is satisfied with constant b being set to $n^{1/2} X_b B_b$. Since the requirements on $\{\tilde{h}_j\}_j$ and $\{\tilde{\beta}_j\}_j$ are satisfied, we want to choose the right value of positive integer K such that Lemma 2.5.8 is satisfied and also we would like the squared distance between y and y_K to be less than $n\epsilon^2$. To do this, we pick K to be the bigger of the two quantities: $X_b^2 B_b^2 / \epsilon^2$ and that given in Lemma 2.5.8. That is,

$$K = \left\lceil \max \left\{ \frac{X_b^2 B_b^2}{\epsilon^2}, \frac{n X_b^2 B_b^2}{\left[\min_{\nu=1,...,V} \frac{\delta_\nu}{\sum_{j=1}^p |\tilde{c}_{j\nu}|} \right]^2 \lambda_{\min}({X_{sL}}^T X_{sL})} \right\} \right\rceil. \tag{2.23}$$

This will force our discretization for the cover to be sufficiently fine that things will work out: we will be able to count the number of cover points in our finite set, and that will be our covering number.

To summarize, with this choice, for any $y \in \mathcal{F}_{|S}$, we can find integers $k_1, ..., k_p$ such that the following hold simultaneously:

a. (It gives a valid discretization of y.) $\sum_{i=1}^p |k_i| \leq K$,

b. (It gives a good approximation to y.) The approximation $y_K = \sum_{j=1}^p \frac{k_i}{K} \tilde{h}_j$ is $\epsilon\sqrt{n}$ close to $y = \sum_{j=1}^p \tilde{\beta}_j \tilde{h}_j$. That is,

$$\|y - y_K\|_2^2 \leq \frac{n X_b^2 B_b^2}{K} \leq n\epsilon^2, \text{and}$$

c. (It obeys operational cost constraints.) $\sum_{j=1}^{p} \tilde{c}_{j\nu}\frac{k_j}{K} \leq 1,\ \nu = 1, ..., V.$

In the above, the existence of $k_1, ..., k_p$ satisfying (a) and (b) comes from Lemma 2.5.7 where we have also used K satisfying $K \geq X_b^2 B_b^2/\epsilon^2 \geq 1$. Lemma 2.5.8 along with the choice of K from (2.23) guarantees that (c) holds as well for this choice of $k_1, ..., k_p$.

Thus, by (b), any $y \in \mathcal{F}_{|S}$ is within $\epsilon\sqrt{n}$ in ℓ_2 distance of at least one of the vectors with coefficients $k_1/K, ..., k_p/K$. Therefore counting the number of p-tuple integers $k_1, ..., k_p$ such that (a) and (c) hold, or equivalently the number of solutions to (2.15), gives a bound on the covering number, which is $|P_c^K|$. That is,

$$N(\sqrt{n}\epsilon, \mathcal{F}_{|S}, \|\cdot\|_2) \leq |P_c^K|.$$

If we did not have any linear constraints, we would have the following bound,

$$N(\sqrt{n}\epsilon, \mathcal{F}_{|S}, \|\cdot\|_2) \leq |P^{K_0}|,$$

where $K_0 := \left\lceil \frac{X_b^2 B_b^2}{\epsilon^2} \right\rceil$ by using Lemma 2.5.7 and very similar arguments as above.

In addition, when $\epsilon \geq X_b B_b$, the covering number is exactly equal to 1 since we can cover the set $\mathcal{F}_{|S}$ by a closed ball of radius $\sqrt{n} X_b B_b$.

Thus we modify our upper bound by taking the minimum of the two quantities $|P^{K_0}|$ and $|P_c^K|$ appropriately to get the result:

$$N(\sqrt{n}\epsilon, \mathcal{F}_{|S}, \|\cdot\|_2) \leq \begin{cases} \min\{|P^{K_0}|, |P_c^K|\} & \text{if } \epsilon < X_b B_b \\ 1 & \text{otherwise.} \end{cases}$$

□

Since Theorem 2.5.1 suggests that $|P_c^K|$ may be an important quantity for the learning process, we discuss how to compute it. We assume that $\tilde{c}_{j\nu}$ are rationals for all $j = 1, .., p, \nu = 1, ..., V$, so that we can multiply each of the V constraints describing P_c^K by the corresponding gcd of the p denominators. This is without loss of generality because the rationals are dense in the reals. This ensures that all the constraints describing polyhedron P_c^K have integer coefficients. Once this is achieved, we can run Barvinok's algorithm [using for example, Lattice Point Enumeration, see De Loera, 2005, and references therein] that counts integer points inside polyhedra and runs in polynomial time for fixed dimension (which is p here). Using the output of this algorithm within our generalization bound will yield a much tighter bound than in previous works [for example, the bound in Zhang, 2002, Theorem 3], especially when $(r, q) = (\infty, 1)$; this is true simply because we are counting more carefully. Note that counting integer points in polyhedrons is a fundamental question in a variety of fields including number theory, discrete optimization, combinatorics to name a few, and making an explicit connection to bounds on the covering number for linear function classes can potentially open doors for better sample complexity bounds.

2.6 Discussion and Conclusion

The perspective taken in this work contrasts with traditional decision analysis and predictive modeling; in these fields, a single decision is often the only end goal. Our goal involves exploring how predictive modeling influences decisions and their costs. Unlike traditional predictive modeling, our regularization terms involve optimization problems, and are not the usual vector norms.

The simultaneous process serves as a way to understand uncertainty in decision-making, and can be directly applied to real problems. We centered our discussion and demonstrations around three questions, namely: "What is a reasonable amount to allocate for this task so we can react best to whatever nature brings?" (answered in Section 2.3), "Can we produce a reasonable probabilistic model, supported by data, where we might expect to pay a specific amount?" (answered in Section 2.3), and "Can our intuition about how much it will cost to solve a problem help us produce a better probabilistic model?" (answered in Section 2.5). The first two were answered by exploring how optimistic and pessimistic views can influence the probabilistic models and the operational cost range. Given the range of reasonable costs, we could allocate resources effectively for whatever nature brings. Also given a specific cost value, we could pick a corresponding probabilistic model and verify that it can be supported by data. The third question was comprehensively answered in Section 2.5 by evaluating how intuition about the operational cost can restrict the probabilistic model space and in turn lead to better sample complexity if the intuition is correct. These are questions that are not handled in a natural way by current paradigms. Answering these three questions are not the only uses for the simultaneous process. For instance, domain experts could use the simultaneous process to explore the space of probabilistic models and policies, and then simply pick the policy among these that most agrees with their intuition. Or, they could use the method to refine the probabilistic model, in order to exclude solutions that the simultaneous process found that did not agree with their intuition.

The simultaneous process is useful in cases where there are many potentially good probabilistic models, yielding a large number of (optimal-response) policies. This happens when the training data are scarce, or the dimensionality of the problem is large compared to the sample size, and the operational cost is not smooth. These conditions are not difficult to satisfy, and do occur commonly. For instance, data can be scarce (relative to the number of features) when they are expensive to collect, or when each each instance represents a real-world entity where few exist; for instance, each example might be a product, customer, purchase record, or historic event. Operational cost calculations commonly involve discrete optimization; there can be many scheduling, knapsack, routing, constraint-satisfaction, facility location, and matching problems, well beyond what we considered in our simple examples. The simultaneous process can be used in cases where the optimization problem is difficult enough that sampling the posterior of Bayesian models, with computing the policy at each round, is not feasible.

We end the chapter by discussing the applicability of our policy-oriented estimation strategy in the real world. Prediction is the end goal for machine learning problems in vision, image processing and biology, and in other scientific domains, but there are many domains where the learning algorithm is used to make recommendations for a subsequent task. We showed applications in Section 2.3 but it is not hard to find applications in other domains, where using either the traditional sequential process, decision theory, or robust optimization may not suffice. Here are some other potential domains:

- Internet advertising, where the goal of the advertising platform is to choose which ad to show a customer. For each customer and advertiser, there is an uncertain estimate of the probability that the customer will click the ad from that advertiser. These estimates determine which ad will be shown next, which is a discrete decision [Muthukrishnan et al., 2007].

- Portfolio management, where we allocate our budget among n risky assets with uncertain returns, and each asset has a different cost associated with the investment [Konno and Yamazaki, 1991].

- Maintenance applications [in addition to the ML&TRP Tulabandhula et al., 2011], where we estimate probabilities of failure for each piece of equipment, and create a policy for repairing, inspecting, or replacing the equipment. Certain repairs are more expensive than others, so the costs of various policy decisions could potentially change steeply as the probability model changes.

- Traffic flows on transportation networks, where the problem can be that of load balancing based on resource constraints and forecasted demands [Koulakezian et al., 2012].

- Policy decisions based on dynamical system simulations, for instance, climate policy, where a politician wants to understand the uncertainty in policy decisions based on the results of a large-scale simulation. If the simulation cannot be computed for all initial values, its result can be estimated using a machine learning algorithm [Barton et al., 2010].

- Pharmaceutical companies choosing a subset of possible drug targets to test, where the drugs are predicted to be effective, and cannot be overly expensive to produce [Yu et al., 2012]. This might be similar in many ways to the real-estate purchasing problem discussed in Section 2.3.

- Machine task scheduling on multi-core processors, where we need to allocate processors to various jobs during a large computation. This could be very similar to the problem of scheduling with constraints addressed in Section 2.3. If we optimistically estimate the amount of time each job takes, we will hopefully free up processors on time so they can be ready for the next part of the computation.

We believe the simultaneous process will open the door for other methods dealing with the interaction of machine learning and decision-making that fall outside the realm of the usual paradigms.

Chapter 3

On Combining Machine Learning with Decision Making

3.1 Introduction

In many domains, it is essential to understand how uncertainty in predictions influences decision-making. In that sense, one would like to explore the space of possible reasonable predictions and understand the range of reasonable policies and their costs. The new framework of Machine Learning with Operational Costs (MLOC) [Tulabandhula and Rudin, 2013] provides a mechanism to do this, and is a type of exploratory decision theory. Where usual decision theories provide a single policy that minimizes expected costs, the MLOC framework is able to produce a range of reasonable policies that span the full set of reasonable costs. To do this, the operational cost becomes a regularization term within the machine learning model, and adjusting the regularization constant allows us to explore solutions for all reasonable costs. This gives decision makers a way to understand the uncertainty in their predictive model in terms of something they can grasp - uncertainty in the cost to solve the problem.

The MLOC framework can also be used in another way, namely to incorporate prior knowledge about the cost to produce a better predictive model. In that sense, knowledge about the cost translates into a more restricted hypothesis space, which potentially translates into better generalization. In particular, if the hypothesis space is restricted, then upper bounds on the complexity of the hypothesis space are smaller, leading to better generalization bounds.

In this work, we provide an application of the MLOC framework to power grid engineering and reliability. This problem, called the *Machine Learning and Traveling Repairman Problem* (ML&TRP), has a machine learning component and a decision-making component. The machine learning component is to predict future power grid failures before they occur, where these failures occur at equipment that is distributed throughout the city. The decision-making component is to determine in what order the equipment should be inspected. We could use the MLOC framework in either of the two ways outlined above: either to understand the range of reasonable costs for the power company, or to use prior knowledge that the costs are high or low in order to choose a more predictive and cost-effective route.

To be more precise, the ML&TRP *prediction* problem is to determine the failure probability for each node on a graph, using features of each node and past failure data. The *decision* problem is to determine a route for a "repair crew" on the graph, where there is some travel time between each pair of nodes. There are many possible applications of the ML&TRP, including the scheduling of safety inspections or repair work for the electrical grid, oil rigs, underground mining, machines in a factory, or airlines. In our experiments, we use data from an ongoing project with Con Edison, which is NYC's power utility company.

We also provide a generalization bound for the MLOC framework based on covering numbers. These bounds are different than those of Tulabandhula and Rudin [2013] which use concentration of Rademacher complexity and Dudley's entropy integral, and are not directly comparable. The bounds here have a much more geometric flavor looking at the hypothesis space as a volumetric object. Neither of the two bounds are tighter in all situations. We find the bounds here to be more intuitive, as the geometry is more transparent.

The ML&TRP relates to literature on both machine learning and optimization (time-dependent traveling salesman problems). In machine learning, our work bears a slight resemblance to work on graph-based regularization [Agarwal, 2006, Belkin et al., 2006, Zhou et al., 2004], but their goal is to obtain probability estimates that are smoothed on a graph with suitably designed edge weights. On the other hand, our goal is to obtain, in addition to probability estimates, a low-cost route for traversing a very different graph with edge weights that are physical distances. Our regularization is vastly different from popular ones (ℓ_1 or ℓ_2 norm) because our regularization comes from beliefs on decision-making costs. We use unlabeled data as does semi-supervised learning [Chapelle et al., 2006] but differ in the motivation as well as the way we use these additional data. For example, we do not extract distributional information from the unlabeled data. Our work contributes to the literature on the TRP (Traveling Repairman Problem) and related problems by adding the new dimension of probabilistic estimation at the nodes. We create new adaptations of modern techniques [Fischetti et al., 1993, Eijl van, 1995, Lechmann, 2009] within our work for solving the TRP part of the ML&TRP.

There is a body of literature regarding cost models for maintenance in the reliability modeling literature, though the emphasis in those works is usually to design a model that accurately represents the stochastic process for the failures. In that literature, for instance, a maintenance schedule would be created from the predicted condition of the equipment (but not on the cost of performing the repairs in a certain order or routing a vehicle between the equipment). Barbera et al. [1996] develop a model that assumes that equipment have exponential rates of failure and fail only once in an inspection interval, and they use this model to determine a maintenance schedule. Marseguerra et al. [2002] introduces a model for degradation leading to failure for a continuous complex system, and use Monte Carlo simulations to determine the optimal degradation level to perform an inspection. Their work uses a very different cost model from ours; the cost is the long run average maintenance cost and cost of failures. A neural-network based maintenance model was developed by Heng et al. [2009]. A related work on routing for emergency maintenance on the electrical grid is the heuristic algorithm of Weintraub et al. [1999] that dispatches vehicles to areas where there are currently breakdowns and where there are likely to be breakdowns in the future. Ertekin et al. [2013] propose a model for failures of power grid equipment and use this model to simulate the cost of various inspection policies.

One can view the MLOC framework to be analogous to a Bayesian approach, in the sense that prior knowledge is being used when not enough data are available.

In Section 3.2 we review the MLOC framework. In Section 3.3 we will motivate and outline the new application of the MLOC framework to the ML&TRP, providing two ways of modeling failure cost. In Section 3.4 we provide mixed-integer nonlinear (MINLP) formulations and discuss algorithms an illustrative example. Section 3.5 gives experimental results on data from the NYC power grid, showing the benefit of the ML&TRP over traditional methods. Section 3.6 contains the theoretical generalization result for the MLOC framework with proofs. Section 3.8 concludes the chapter. The conference paper of [Tulabandhula et al., 2011] contains a summary of work on the ML&TRP, and the paper Tulabandhula and Rudin [2013] provides a more complete explanation of the MLOC framework, with other illustrations and connections to robust optimization.

3.2 Review of Framework for Machine Learning with Operational Costs

In the MLOC framework we have the standard supervised training set of labeled instances, $\{(x_i, y_i)\}_{i=1}^m$, where $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$. For simplicity, $\mathcal{X} \subset \mathbb{R}^d$. To have nonlinear functions, we could simply have the j^{th} component of x replaced by a nonlinear function $h_j(x)$. Also $\mathcal{Y} \subset \mathbb{R}$. We wish to learn a function $f^* : \mathcal{X} \to \mathcal{Y}$. This is ordinarily done by solving a minimization problem:

$$f^* \in \text{argmin}_{f \in \mathcal{F}^{unc}} \left(\sum_{i=1}^{m} l(f(x_i), y_i) + C_2 R(f) \right), \tag{3.1}$$

for some loss function $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$, regularizer $R : \mathcal{F}^{unc} \to \mathbb{R}$, constant C_2 and function class $\mathcal{F}^{unc}$. $\mathcal{F}^{unc}$ is the set of all linear functionals, where $f \in \mathcal{F}^{unc}$ is of the form $\lambda \cdot x$, $\lambda \in \mathbb{R}^d$. The superscript '*unc*' refers to the word "unconstrained."

Consider an organization making a policy decision regarding a new collection of unlabeled instances $\{\tilde{x}_i\}_{i=1}^M \in \mathcal{X}^M$. The cost to enact a policy is not exactly known, because the labels for the $\{\tilde{x}_i\}_i$ are not known. Instead the model's predictions are used, which are the $f^*(\tilde{x}_i)$'s. The goal of the organization is then to create a policy π^* that minimizes operational cost $\text{OpCost}(\pi, f^*, \{\tilde{x}_i\}_i)$. The operational cost $\text{OpCost}(\pi, f^*, \{\tilde{x}_i\}_i)$ is how much will be spent if policy π is chosen in response to the $\{f^*(\tilde{x}_i)\}_i$'s. When there is uncertainty in f^*, there is uncertainty in the cost to enact the optimal policy π^*. This uncertainty is what we would like to explore. A typical way that companies make decisions is using what we call the **sequential process**, which computes the policy according to two steps:

Step 1: Create function f^* based on $\{(x_i, y_i)\}_i$ according to (3.1). That is:

$$f^* \in \text{argmin}_{f \in \mathcal{F}^{unc}} \left(\sum_{i=1}^{m} l(f(x_i), y_i) + C_2 R(f) \right).$$

Step 2: Choose policy π^* to minimize the operational cost,

$$\pi^* \in \text{argmin}_{\pi \in \Pi} \text{OpCost}(\pi, f^*, \{\tilde{x}_i\}_i).$$

On the other hand, the MLOC framework is based around a **simultaneous process**, which combines Steps 1 and 2 of the sequential process. To do this, the operational cost becomes a regularization term, and its regularization parameter C_1 controls the amount of optimism or pessimism for the operational cost.

Step 1: Choose a model f^* obeying the following:

$$f^* \in \underset{f \in \mathcal{F}^{unc}}{\text{argmin}} \left[\sum_{i=1}^{m} l\left(f(x_i), y_i\right) + C_2 R(f) + C_1 \min_{\pi \in \Pi} \text{OpCost}\left(\pi, f, \{\tilde{x}_i\}_i\right) \right].$$

Step 2: Compute the policy:

$$\pi^* \in \underset{\pi \in \Pi}{\text{argmin}}\, \text{OpCost}\left(\pi, f^*, \{\tilde{x}_i\}_i\right).$$

The case $C_1 = 0$ for the simultaneous process is precisely the sequential process; thus, the sequential process is a special case of the simultaneous process. Our ability to solve the MLOC simultaneous process depends on the tractability of the optimization problem $\underset{\pi \in \Pi}{\operatorname{argmin}} \operatorname{OpCost}(\pi, f^*, \{\tilde{x}_i\}_i)$. However, if this problem is intractable, then the sequential process is also intractable, and the organization will not be able to choose an optimized policy at all. The simultaneous process requires this subproblem to be solved several times, whereas the sequential process only requires the subproblem to be solved once. If the number of unlabeled instances is small, then Step 1 can be solved without a problem, even if the training set is large. As C_1 varies over its full range, it maps out the full range of costs for all reasonable solutions. If C_1 is set to a number that is too large (either positive or negative), the solution of the simultaneous process will have empirical error that is too high to be reasonable. In that case, we know that by varying C_1 within a smaller range will lead to the full range of costs for reasonable predictive models.

As with any regularization term, the new operational cost term can be interpreted as a prior belief about the model - in this case, a belief that the operating costs should be lower or higher on the current set of unlabeled instances $\{\tilde{x}_i\}_i$. In that sense, MLOC regularization may have a closer connection to reality than typical (e.g., ℓ_1 or ℓ_2 norm) regularizers. If one asks a manager at a company what prior belief they have about the estimation model, it is not likely they would give a answer in terms of coefficients for a linear model. Even managers who are not mathematicians or computer scientists might have some belief - they could perhaps believe that they are expecting to spend a certain amount to enact the policy. It is possible that this type of belief, which relies on direct experience, might be more practical, and more accurate, than the more abstract prior information that we are typically used to dealing with. In the ML&TRP, the training error term is derived from data from the past, and the OpCost term is calculated on data from the present. The OpCost term is the only term that deals with routing.

3.3 The Machine Learning and Traveling Repairman Problem

The US Department of Energy's Grid 2030 document states that "America's electric system, 'the supreme engineering achievement of the 20th century,' is aging, inefficient, and congested, and incapable of meeting the future energy needs of the Information Economy without operational changes and substantial capital investment over the next several decades" [United States Department of Energy and Distribution, 2003]. Since 2004, many power utility companies are implementing new inspection and repair programs for preemptive maintenance, whereas in the past, all repair work was done reactively [Urbina, 2004]. New York City has the oldest power system in the world, and the largest underground electric system, with enough electrical cable to go three and a half times around the world. In New York City, there are several separate new preemptive maintenance programs, including the targeted inspection program for electrical service structures (manholes), programs that perform extensive repairs that were placed on a waiting list after the manhole was inspected, and the *vented cover replacement program*, where each manhole is replaced with a vented cover that allows gases to escape, mitigating the possibility and effects of serious events including fires and explosions. Con Edison, the power company in NYC, has the ability to use machine learning models in Manhattan, Brooklyn and the Bronx for scheduling of manhole inspection and repair work [Rudin et al., 2010, 2012b, 2011, 2014]. This project was the motivation for the development of the ML&TRP and we use data from the NYC power grid for our experiments. Features for the NYC model are derived from physical characteristics of the manhole (e.g., number of electrical cables entering the manhole), and features derived from its history of involvement in past events. Repeat failures (serious and non-serious events) can occur on the same manhole. We take the possibility of repeat failures into account in the ML&TRP (in Cost 1 given below). That said, failures are rare events, and it is not easy to accurately estimate the probability that a given manhole will fail within a given period of time. Because of this uncertainty, we can use the MLOC framework to assist in decision-making. The result $\pi^* \in \Pi$ from the algorithm would be a route that could be used for the repair crew to fix a pre-specified set of manholes corresponding to $\{\tilde{x}_i\}_{i=1}^{M}$, which are assumed to need a particular repair.

3.3.1 Learning

In what follows, we will use descriptions and terminology that match the power grid application. In the ML&TRP, data from the past will be used to train the model, denoted $\{(x_i, y_i)\}_{i=1}^{m}$, whereas the $\tilde{x}_i$ are calculated from the present, whose labels are from the future and thus not known. Let x_i^j indicate the j-th coordinate of the feature vector for manhole i calculated at a time period from the past. The x_i vector encodes the number and types of electrical cables, number and types of previous events, etc. The label for manhole i from the past is denoted y_i, where $y_i \in \{-1, 1\}$ indicating whether the manhole had a failure (fire, explosion, smoking manhole) within a specific period of time in the past. More details about the features and labels can be found in Section 3.5. The other instances $\{\tilde{x}_i\}_{i=1}^{M}$ (with M unrelated to m), are unlabeled data that are each associated with a node on a graph G. The nodes of the graph G indexed by $i = 1, ..., M$ represent manholes on which we want to design a route. Note that M can be substantially smaller than m, e.g., $M < 10$ and $m > 20,000$; e.g., for a repair truck that carries supplies for at most M repairs. We are also given physical distances $d_{i,j} \in \mathbb{R}_+$ between each pair of nodes i and j. A route on G is represented by a permutation π of the node indices $1, \ldots, M$. Let Π be the set of all permutations of $\{1, ..., M\}$. Failure probabilities will be estimated at each of the nodes and these estimates will be based on a function of the form $f_\lambda(x) = \lambda \cdot x$. The class of possible functions $\mathcal{F}$ is chosen to be: $\mathcal{F} := \{f_\lambda : \lambda \in \mathbb{R}^d, \|\lambda\|_2 \leq B_b\}$, where B_b is a fixed positive real number. We choose the logistic loss: $l(f_\lambda(x), y) := \ln\left(1 + e^{-yf_\lambda(x)}\right)$ so that the probability of failure $P(y = 1|x)$, is estimated as in logistic regression by:

$$P(y = 1|x) \text{ or } p(x) := \frac{1}{1 + e^{-f_\lambda(x)}}. \tag{3.2}$$

Note that the routing problem is done in batch: once the route is determined, the repair truck is sent out and changes to the route are no longer possible.

3.3.2 Two Options for the OpCost

The operational cost can be defined to match the application. In the first option (denoted as Cost 1), for each node there is a cost for (possibly repeated) failures prior to a visit by the repair crew. In this case, temporary repairs are made to fix each node before the repair crew comes to make permanent repairs. In the second option (denoted as Cost 2), for each node, there is a cost for the first failure prior to visiting it. In this case, permanent repairs are made when there is an event, or when the repair crew arrives, whichever is sooner. There is a natural interpretation of the failures as being generated by a continuous random process at each of the nodes. When discretized in time, this is approximated by a Bernoulli process with parameter $p(\tilde{x}_i)$. Both Cost 1 and Cost 2 are appropriate for power grid applications. Cost 2 is also appropriate for delivery truck routing applications, where perishable items can fail (once an item has spoiled, it cannot spoil again).

For convenience, we assume that after the repair crew visits all the nodes, it returns to the starting node (node 1) which is fixed beforehand. Scenarios where one is not interested in beginning from or returning to the starting node would be modeled slightly differently (the computational complexity remains the same). Let a route be represented by $\pi : \{1, ..., M\} \mapsto \{1, ..., M\}$, this means that $\pi(i)$ is the i^{th} node to be visited. For example, let $M = 4, \pi = [2, 3, 4, 1]$. This means, $\pi(1) = 2$, node 2 is the first node to be visited, $\pi(2) = 3$, node 3 is the second node on the route, and so on. Since the final node visited is the first node, we append the following to the definition of π: $\pi(M + 1) = \pi(1)$. Let the distances be scaled appropriately so that a unit of distance is traversed in a unit of time. Given a route, the *latency* of a node $\pi(i)$ is the time (or equivalently distance) from the start at which node $\pi(i)$ is visited. It is the sum of distances traversed before position i on the route:

$$L_\pi(\pi(i)) := \begin{cases} \sum_{k=1}^{M} d_{\pi(k)\pi(k+1)} \mathbf{1}_{[k<i]} & i = 2, ..., M \\ \sum_{k=1}^{M} d_{\pi(k)\pi(k+1)} & i = 1. \end{cases} \tag{3.3}$$

The starting node $\pi(1)$ thus has a latency $L_\pi(\pi(1))$ which is the total length of the route starting at node $\pi(1)$ and ending at node $\pi(1)$ after visiting all other nodes.

Cost 1: Cost is Proportional to Expected Number of Failures Before the Visit

Up to the time that node $\pi(i)$ is visited by the repair crew, there is a probability $p(\tilde{x}_{\pi(i)})$ that a failure will occur within each unit time interval. Equivalently, within each unit time interval, failures are determined by a Bernoulli random variable with parameter $p(\tilde{x}_{\pi(i)})$. Thus, in a time interval of length $L_\pi(\pi(i))$ units, the number of node failures follows the binomial distribution Bin $\left(L_\pi(\pi(i)), p(\tilde{x}_{\pi(i)})\right)$. For each node, we will associate a cost proportional to the expected number of failures before the repair crew's visit, as follows:

$$\begin{aligned}\text{Cost of node } \pi(i) &\propto E(\text{number failures in } L_\pi(\pi(i)) \text{ time units}) \\ &= \text{mean of Bin}(L_\pi(\pi(i)), p(\tilde{x}_{\pi(i)})) = p(\tilde{x}_{\pi(i)})L_\pi(\pi(i)). \quad (3.4)\end{aligned}$$

Using this cost, if the failure probability for node $\pi(i)$ is small, we can afford to visit it later on, trading off its latency $L_\pi(\pi(i))$. If $p(\tilde{x}_{\pi(i)})$ is large, we should visit node $\pi(i)$ earlier to keep our overall failure cost low. The failure cost of route π is then $\text{OpCost}(\pi, f_\lambda, \{\tilde{x}_i\}_{i=1}^M, \{d_{i,j}\}_{i,j=1}^M) = \sum_{i=1}^M p(\tilde{x}_{\pi(i)})L_\pi(\pi(i))$.

Substituting the definition of $L_\pi(\pi(i))$ from (3.3):

$$\begin{aligned}&\text{OpCost}(\pi, f_\lambda, \{\tilde{x}_i\}_{i=1}^M, \{d_{i,j}\}_{i,j=1}^M) = \\ &\quad \sum_{i=2}^M p(\tilde{x}_{\pi(i)}) \sum_{k=1}^M d_{\pi(k)\pi(k+1)} 1_{[k<i]} + p(\tilde{x}_{\pi(1)}) \sum_{k=1}^M d_{\pi(k)\pi(k+1)}, \qquad (3.5)\end{aligned}$$

where $p(\tilde{x}_{\pi(i)})$ is given in (3.2). This will be Cost 1. There are ways to make Cost 1 more general. The individual node cost in (3.4) assumes that the node's failure probability $p(\tilde{x}_{\pi(i)})$ becomes zero after the repair crew's visit, so that for the remainder of the route, the cost incurred at this node is $\propto 0 \times (L_\pi(\pi(1)) - L_\pi(\pi(i)))$. We could relax this by assuming $p(\tilde{x}_{\pi(i)})$ does not vanish after the repair crew's visit and adding an additional cost for the expected failures in this period. That is, if β is a constant of proportionality for the cost after visiting node $\pi(i)$, then the cost would become:

$$\text{Cost of node } \pi(i) = \beta\left[L_\pi(\pi(1)) - L_\pi(\pi(i))\right] p(\tilde{x}_{\pi(i)}) + L_\pi(\pi(i)) p(\tilde{x}_{\pi(i)}).$$

If $\beta = 1$, then the repair crew does not have any effect and cost of each node is independent of its expected number of failures before the repair crew's visit. Typically, we expect that the repair crew will repair the node so that it will not fail, and the second term above is much larger than the first. Taking the constant of proportionality as $\beta = 0$, we return to the individual costs given by (3.4).

Note that since the cost is a sum of M terms, it is invariant to ordering or indexing (caused by π). Thus we can rewrite the cost as

$$\text{OpCost}(\pi, f_\lambda, \{\tilde{x}_i\}_{i=1}^M, \{d_{i,j}\}_{i,j=1}^M) = \sum_{i=1}^M p(\tilde{x}_i) L_\pi(i). \tag{3.6}$$

Cost 2: Cost is Proportional to Probability that the First Failure is Before the Visit

This cost reflects the penalty for not visiting a node before the first failure occurs there. This model is governed by the geometric distribution. Let the parameter of the distribution be p. Then the probability that the first failure for node $\pi(i)$ occurs at time index $t > 0$ is $p(1-p)^{t-1}$. The probability that the first failure for node $\pi(i)$ occurs before time $L_\pi(\pi(i))$ is then the sum of the failure probabilities from $t = 1, ..., L_\pi(\pi(i)) : \sum_{t=1}^{L_\pi(\pi(i))} p(1-p)^{t-1} = 1 - (1-p)^{L_\pi(\pi(i))}$. Thus, substituting the expression (3.2) for p, we have:

$$\begin{aligned} &P\Big(\text{first failure occurs before time } L_\pi(\pi(i))\Big) = 1 - (1 - p(\tilde{x}_{\pi(i)}))^{L_\pi(\pi(i))} \\ = &\ 1 - \left(1 - \frac{1}{1 + e^{-f_\lambda(\tilde{x}_{\pi(i)})}}\right)^{L_\pi(\pi(i))} = 1 - \left(1 + e^{f_\lambda(\tilde{x}_{\pi(i)})}\right)^{-L_\pi(\pi(i))}. \end{aligned}$$

The cost of visiting node $\pi(i)$ will be proportional to this quantity:

$$\text{Cost of node } \pi(i) \quad \propto \quad \left(1 - \left(1 + e^{f_\lambda(\tilde{x}_{\pi(i)})}\right)^{-L_\pi(\pi(i))}\right). \tag{3.7}$$

Similarly to Cost 1, $L_\pi(\pi(i))$ influences the cost at each node. If we visit a node early in the route, then the cost incurred is small because the node is less likely to fail before we reach it. Similarly, if we schedule a visit later on in the tour, the cost is higher because the node has a higher chance of failing prior to the repair crew's visit.

The total failure cost is thus:

$$\text{OpCost}(\pi, f_\lambda, \{\tilde{x}_i\}_{i=1}^M, \{d_{i,j}\}_{i,j=1}^M) = \sum_{i=1}^{M} \left(1 - \left(1 + e^{f_\lambda(\tilde{x}_{\pi(i)})}\right)^{-L_\pi(\pi(i))}\right). \tag{3.8}$$

This cost is not directly related to a weighted TRP cost in its present form. That is, when the failure probabilities of the nodes are all the same, the total cost is not linear in the latencies, as is the case for Cost 1. Building on this cost, we will derive a cost that is the same as a weighted TRP in Section 3.4.2, of the form:

$$\text{Cost of node } \pi(i) \quad \propto \quad L_\pi(\pi(i)) \log\left(1 + e^{f_\lambda(\tilde{x}_{\pi(i)})}\right), \tag{3.9}$$

as an alternative to (3.7).

There is a slightly more general version of this formulation (as there was for Cost 1), which is to take the cost for each node to be a function of two quantities: the probability of failure before the visit, and the probability of failure after the visit. Let us redefine β to be a constant of proportionality for the cost of visiting before the failure event. From the geometric distribution, P(failure occurs after time $L_\pi(\pi(i))) = (1 - p(\tilde{x}_{\pi(i)}))^{L_\pi(\pi(i))}$, and the cost of visiting node $\pi(i)$ becomes:

$$\text{Cost of node } \pi(i) \quad \propto \quad P(\text{failure before } L_\pi(\pi(i))) + \beta \times P\left(\text{failure after } L_\pi(\pi(i))\right).$$

If $\beta = 1$, then the sum above is 1 for all nodes regardless of node failures or latencies.

More realistically, the cost of visiting the node after the failure is more than the cost of visiting proactively, $\beta \ll 1$ leading to (3.7). We could again have written the summation to hide the dependence on π:

$$\text{OpCost}(\pi, f_\lambda, \{\tilde{x}_i\}_{i=1}^M, \{d_{i,j}\}_{i,j=1}^M) = \sum_{i=1}^{M} \left(1 - \left(1 + e^{f_\lambda(\tilde{x}_i)}\right)^{-L_\pi(i)}\right).$$

Remark 3.3.1. The costs defined above are by no means exhaustive. We chose to define operational costs this way because they mimic the well known minimum latency objective in routing problems. For instance, we could have used a Poisson failure model at each node instead of binomial or geometric as in Costs 1 and 2. Let us assume that the Poisson rate parameter $\mu(\tilde{x}_{\pi(i)})$ is the output of the estimation problem (say proportional to $p(\tilde{x}_{\pi(i)})$). Then

$$P(k \text{ failures occur in time } L_\pi(\pi(i))) = \frac{(\mu(\tilde{x}_{\pi(i)})L_\pi(\pi(i)))^k e^{-\mu(\tilde{x}_{\pi(i)})L_\pi(\pi(i))}}{k!}.$$

From this we can get the probability that at least one failure occurs in time interval $[0, L_\pi(\pi(i))]$ at node $\pi(i)$. Now we can define the operational cost to be the sum of these probabilities which depend on the routing and proceed in the same way as Cost 2. That is, we can minimize this cost to get the optimal routing π^*.

Remark 3.3.2. The operational cost must depend on graph properties like latency. We would not like to minimize an objective of the form $\sum_{i=1}^M \frac{1}{p(\tilde{x}_{\pi(i)})}$ (or any other function of just $p(\tilde{x}_{\pi(i)})$, the output of the estimation problem) as this does not lead to an operational cost in the true sense. This operational cost does not make use of latency information or other graph properties related to routing unless $p(\tilde{x}_{\pi(i)})$ implicitly depends on them (which is not the case here).

Now that the major steps for both formulations have been defined, we will discuss methods for optimizing the objectives.

3.4 Optimization

We start by formulating mixed-integer linear programs (MILP's) for the TRP subproblem.

3.4.1 Mixed-integer optimization for Cost 1

For either the sequential or simultaneous processes, we need the solution of the subproblem: $\pi^* \in \text{argmin}_{\pi\in\Pi}\text{OpCost}(\pi, f_\lambda^*, \{\tilde{x}_i\}_{i=1}^M, \{d_{i,j}\}_{i,j=1}^M)$, or equivalently,

$$\pi^* \;\in\; \arg\min_{\pi\in\Pi} \sum_{i=2}^{M} p(\tilde{x}_{\pi(i)}) \sum_{k=1}^{M} d_{\pi(k)\pi(k+1)} \mathbf{1}_{[k<i]} + p(\tilde{x}_{\pi(1)}) \sum_{k=1}^{M} d_{\pi(k)\pi(k+1)}. \quad (3.10)$$

Let us compare this to the standard traveling repairman problem (TRP) problem [see Blum et al., 1994]:

$$\pi^* \in \text{argmin}_{\pi\in\Pi} \sum_{k=1}^{M} d_{\pi(k)\pi(k+1)}(M+1-k). \quad (3.11)$$

The standard TRP objective (3.11) is a special case of the weighted TRP (3.10) when $\forall i = 1, ..., M,\; p(\tilde{x}_i) = p$:

$$\begin{aligned} &\sum_{i=2}^{M} p(\tilde{x}_{\pi(i)}) \sum_{k=1}^{M} d_{\pi(k)\pi(k+1)} \mathbf{1}_{[k<i]} + p(\tilde{x}_{\pi(1)}) \sum_{k=1}^{M} d_{\pi(k)\pi(k+1)} \\ &= \; p \sum_{i=2}^{M} \sum_{k=1}^{M} d_{\pi(k)\pi(k+1)} \mathbf{1}_{[k<i]} + p \sum_{k=1}^{M} d_{\pi(k)\pi(k+1)} \\ &= \; p \sum_{i=2}^{M} \sum_{k=1}^{M} d_{\pi(k)\pi(k+1)} \mathbf{1}_{[k<i]} + p \sum_{k=1}^{M} d_{\pi(k)\pi(k+1)} \mathbf{1}_{[k<M+1]} \\ &= \; p \sum_{k=1}^{M} d_{\pi(k)\pi(k+1)} \sum_{i=2}^{M+1} \mathbf{1}_{[k<i]} = p \sum_{k=1}^{M} d_{\pi(k)\pi(k+1)}(M+1-k). \end{aligned}$$

The TRP is different from the traveling salesman problem (TSP); the goal of the traveling salesman problem is to minimize the total traversal time (in this case, this is the same as the distance traveled) needed to visit all nodes once, whereas the goal of the traveling repairman problem is to minimize the sum of the waiting times to visit each node. Both the TSP and the TRP are known to be NP-complete in the general case [Blum et al., 1994]. Intuitively, a TRP route cost objective captures the total waiting cost of a service system from the customer's (the node's) point of view. For example, consider a truck carrying prioritized items to be delivered to customers. At each customer's stop, that customer's item is removed from the truck. The goal of the TRP is to minimize the total waiting time of these customers.

We start by extending an integer programming formulation of standard TRP [Fischetti et al., 1993] to include "unequal flow values" so that we can solve (3.10) [there are many other integer programming formulations in the literature as well, see for instance Méndez-Díaz et al., 2008]. The weights $\{\bar{p}(\tilde{x}_i)\}_i$ within the formulation below will be defined later. For interpretation, consider the sum of the probabilities $\sum_{i=1}^{M} \bar{p}(\tilde{x}_i)$ as the total "flow" through a route. At the beginning of the tour, the repair crew has flow $\sum_{i=1}^{M} \bar{p}(\tilde{x}_i)$. Along the tour, flow of the amount $\bar{p}(\tilde{x}_i)$ is dropped when the repair crew visits node $\pi(i)$ at latency $L_\pi(\pi(i))$. In this way, the amount of flow during the tour is the sum of the probabilities $\bar{p}(\tilde{x}_i)$ for nodes that the repair crew has not yet visited. We introduce two sets of variables $\{z_{i,j}\}_{i,j}$ and $\{y_{i,j}\}_{i,j}$ that together represent a route (instead of the π notation). Let $z_{i,j}$ represent the flow on edge (i, j) and let a binary variable $y_{i,j}$ represent whether there exists a flow on edge (i, j). (There will only be a flow along the route, and there will not be a flow along edges that are not in the route.) The mixed-integer program is as follows:

$$\min_{z,y} \sum_{i=1}^{M}\sum_{j=1}^{M} d_{i,j} z_{i,j} \quad \text{s.t.} \tag{3.12}$$

$$\text{No flow from node } i \text{ to itself: } z_{i,i} = 0 \;\; \forall i = 1, ..., M \tag{3.13}$$

$$\text{No edge from node } i \text{ to itself: } y_{i,i} = 0 \;\; \forall i = 1, ..., M \tag{3.14}$$

$$\text{Exactly one edge into each node: } \sum_{i=1}^{M} y_{i,j} = 1 \;\; \forall j = 1, ..., M \tag{3.15}$$

$$\text{Exactly one edge out from each node: } \sum_{j=1}^{M} y_{i,j} = 1 \;\; \forall i = 1, ..., M \tag{3.16}$$

$$\text{Flow coming back to initial point at the end of loop: } \sum_{i=1}^{M} z_{i,1} = \bar{p}(\tilde{x}_1) \qquad (3.17)$$

$$\text{Change of flow after crossing node } k\text{:}$$

$$\sum_{i=1}^{M} z_{i,k} - \sum_{j=1}^{M} z_{k,j} = \begin{cases} \bar{p}(\tilde{x}_1) - \sum_{i=1}^{M} \bar{p}(\tilde{x}_i) & k = 1 \\ \bar{p}(\tilde{x}_k) & k = 2, ..., M \end{cases} \qquad (3.18)$$

$$\text{Connects flows } z \text{ to indicators of edge } y\text{: } z_{i,j} \leq r_{i,j} y_{i,j} \qquad (3.19)$$

$$\text{where } r_{i,j} = \begin{cases} \bar{p}(\tilde{x}_1) & j = 1 \\ \sum_{i=1}^{M} \bar{p}(\tilde{x}_i) & i = 1 \\ \sum_{i=2}^{M} \bar{p}(\tilde{x}_i) & \text{otherwise.} \end{cases}$$

Constraints (3.13) and (3.14) restrict self-loops from forming. Constraints (3.15) and (3.16) ensure that every node should have exactly one edge coming in and one going out. Constraint (3.17) represents the flow on the last edge coming back to the starting node. Constraint (3.18) quantifies the flow change after traversing a node k. Constraint (3.19) represents an upper bound on $z_{i,j}$ relating it to the corresponding binary variable $y_{i,j}$. We can define the weights $\bar{p}(\tilde{x}_i)$, for example, for Cost 1, to be equal to the estimated failure probabilities $1/(1 + e^{-\lambda \cdot \tilde{x}_i})$.

3.4.2 Mixed integer optimization for Cost 2

Here we reason about the choice for changing the cost per node in (3.7) to resemble (3.9). Starting with the sum (3.8) over node costs (3.7), we apply the log function to the second term of the cost of each node (3.7) to get a new cost $\left(1 - \log\left(1 + e^{f_\lambda(\tilde{x}_{\pi(i)})}\right)^{-L_\pi(\pi(i))}\right)$, and the new minimization problem is:

$$\begin{aligned} \min_{\pi} \ & \sum_{i=1}^{M} \left(1 - \log\left(1 + e^{f_\lambda(\tilde{x}_{\pi(i)})}\right)^{-L_\pi(\pi(i))}\right) \\ = \ & -\max_{\pi} \left(\sum_{i=1}^{M} \log\left(1 + e^{f_\lambda(\tilde{x}_{\pi(i)})}\right)^{-L_\pi(\pi(i))} - \text{const} \right) \\ = \ & \min_{\pi} \left[\sum_{i=1}^{M} L_\pi(\pi(i)) \log\left(1 + e^{f_\lambda(\tilde{x}_{\pi(i)})}\right) \right] + \text{const}, \end{aligned}$$

where the first term is the sum over nodes of the expression (3.9). This failure cost term is now a weighted sum of latencies where the weights are of the form $\log\left(1 + e^{f_\lambda(\tilde{x}_{\pi(i)})}\right)$. We can thus reuse the mixed integer program (3.12)-(3.19) where the weights are redefined as $\bar{p}(\tilde{x}_i) := \log\left(1 + e^{\lambda \cdot \tilde{x}_i}\right)$.

Our choices for the cost and failure models above allow us to use a weighted version of the intuitive minimum latency or TRP problem for routing. In particular, the log transformation of individual terms in the original version of Cost 2, (3.8), precisely serves this purpose. In general, depending on the way we define the operational cost and the failure model, they may not necessarily map back to popular routing problems like the TRP as we have here. Nonetheless, there are many valid approaches beyond what we pursue this in this work.

Now that the TRP subproblem has been completely defined for both Cost 1 and Cost 2, we will discuss first how to solve the subproblem alone, which is Step 2 of the sequential process. Then we will discuss the solvers for the simultaneous process.

3.4.3 Solving the weighted TRP subproblem

A generic MILP solver like CPLEX[1] or Gurobi[2] can produce an exact solution using branch-and-bound or other related exact methods. We use Gurobi. The weighted TRP problem is NP-hard (can be shown by a reduction to the Hamiltonian cycle problem) and hence most likely not solvable by polynomial-time algorithms. The standard unweighted (all weights equal) TRP can be encoded by different mixed-integer programming formulations [see Fischetti et al., 1993, Eijl van, 1995, Méndez-Díaz et al., 2008] each with different performance guarantees (e.g., solving 15-60 nodes), which could be adapted for our purpose. There are also techniques for producing constant factor approximate solutions to the unweighted TRP [Goemans and Kleinberg, 1998, Blum et al., 1994, Arora and Karakostas, 2006, Archer et al., 2008, Archer and Blasiak, 2010], which could run faster than the MILP solvers for large problems. If the weights $\{w_i\}_i$ are integers, we can adapt these faster techniques for the standard problem to

[1] IBM ILOG CPLEX Optimization Studio v12.2.0.2 2010

[2] Gurobi Optimizer v3.0, Gurobi Optimization, Inc. 2010

the weighed TRP problem by replicating each node w_i times. If the weights are rational, as is the case in (3.20) and (3.21), we can use rounding and discretization in order to apply the faster solution techniques for solving the standard TRP.

3.4.4 Solving Mixed-integer nonlinear programs (MINLPs)

For the simultaneous process, the inputs to the program are training data $\{x_i, y_i\}_{i=1}^{m}$, unlabeled nodes $\{\tilde{x}_i\}_{i=1}^{M}$ the distances between them $\{d_{i,j}\}_{i,j=1}^{M}$ and constants C_1 and C_2. The full simultaneous process formulation using Cost 1 is:

$$\min_{\lambda}\left(\sum_{i=1}^{m}\ln\left(1+e^{-y_i f_\lambda(x_i)}\right)+C_2\|\lambda\|_2^2+C_1\min_{\{z_{i,j},y_{i,j}\}}\sum_{i=1}^{M}\sum_{j=1}^{M}d_{i,j}z_{i,j}\right) \tag{3.20}$$

$$\text{subject to constraints (3.13) to (3.19), where } \bar{p}(\tilde{x}_i)=\frac{1}{1+e^{-\lambda\cdot\tilde{x}_i}}.$$

The full formulation using the modified version of Cost 2 is:

$$\min_{\lambda}\left(\sum_{i=1}^{m}\ln\left(1+e^{-y_i f_\lambda(x_i)}\right)+C_2\|\lambda\|_2^2+C_1\min_{\{z_{i,j},y_{i,j}\}}\sum_{i=1}^{M}\sum_{j=1}^{M}d_{i,j}z_{i,j}\right) \tag{3.21}$$

$$\text{subject to constraints (3.13) to (3.19) hold, where } \bar{p}(\tilde{x}_i)=\log\left(1+e^{\lambda\cdot\tilde{x}_i}\right).$$

If we have an algorithm for solving (3.20), then the same scheme can be used to solve (3.21). There are multiple ways of solving (or approximately solving) a mixed integer nonlinear optimization problem of the form (3.20) or (3.21). We consider three methods in this work for solving (3.20) and (3.21).

- Generic mixed integer non-linear programming (MINLP) solver (Bonmin).
- Nelder-Mead (NM) which is a iterative scheme over the λ parameter space, solving a weighted TRP subproblem in each iteration.
- Alternating Minimization (AM) which alternatively minimizes over λ and π optimization variables.

Method 1: MINLP Solver

For our experiments we directly use a MINLP solver called Bonmin [Bonami et al., 2008]. These types of solvers typically use general MILP solving techniques like branch and bound or dynamic programming interleaved with continuous optimization. Since the general MILP solving techniques, as discussed, can take exponential time when applied directly to our formulations, the MINLP solvers which use them can in turn, be inefficient if the graph is moderate to large in size. However, when the graph is small, for instance when we want to schedule a tour over only a few nodes, the MINLP solver can generally compute a solution to the problems (3.20) or (3.21) in a manageable period of time.

Method 2: Nelder-Mead in λ-space (NM)

The Nelder-Mead minimization algorithm requires only function evaluations [Nelder and Mead, 1965]. The ML&TRP can be viewed as a minimization in the space of all λ vectors; since we have solvers for the weighted TRP subproblem, we are able to evaluate the ML&TRP objective for a given value of λ. In our experiments we use the MILP solver (Gurobi) for the subproblem. Note that the ML&TRP objective can have non-differentiable kinks arising from discontinuities in the failure cost term; a method that relies on the gradient or Hessian of the objective function might get stuck in narrow local minima, whereas methods that use only function evaluations may not have this problem. The generic Nelder-Mead scheme can have disadvantages with respect to performance [Rios, 2009], in which case, other schemes like Multilevel Coordinated Search (MCS) [Huyer and Neumaier, 1999] can be used in place of Nelder-Mead. Note that since the objective is non-convex, all solutions obtained by NM are only guaranteed to be locally optimal.

Algorithm 1 AM: Alternating minimization algorithm

```
Inputs: $\{x_i, y_i\}_1^m, \{\tilde{x}_i\}_1^M, \{d_{ij}\}_{ij}, C_1, C_2, T$ and initial vector $\lambda_0$.
for t=1:T do
  Compute $\pi_t \in \text{argmin}_{\pi \in \Pi} \text{Obj}(\lambda_{t-1}, \pi)$.
  Compute $\lambda_t \in \text{argmin}_{\lambda \in \mathbf{R}^d} \text{Obj}(\lambda, \pi_t)$.
end for
Output: $\pi_T$.
```

Method 3: Alternating minimization in λ-π space (AM)

Our alternating minimization scheme also operates in the λ-π space as follows. Define the objective Obj as a function of λ and π:

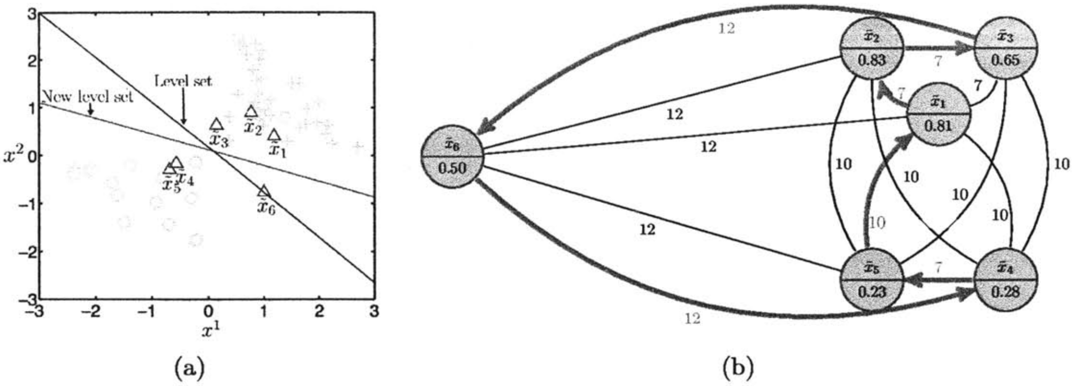
$$\text{Obj}(\lambda, \pi) = \sum_{i=1}^{m} \ln\left(1 + e^{-y_i f_\lambda(x_i)}\right) + C_2 \|\lambda\|_2^2 + C_1 \text{OpCost}\left(\pi, f_\lambda, \{\tilde{x}_i\}_{i=1}^M, \{d_{i,j}\}_{i,j=1}^M\right).$$

Starting from an initial vector λ_0, Obj is minimized alternately with respect to λ and then with respect to π, as shown in Algorithm 1. The second step, solving for π, is the same as solving the TRP subproblem, and we again use the MILP solver for this. Conditions for convergence and correctness for such iterative schemes are given by Csiszár and Tusnády [1984]; again, it is not possible to guarantee globally optimal solutions using this method.

3.4.5 Illustrative Experiment

We will use the ML&TRP to show the fundamental property motivating the MLOC framework: that a large change in the probability model does not necessarily lead to a large change in overall prediction accuracy, but may lead to very different solutions.

The training set was chosen uniformly at random from a distribution that is uniform over two triangles pointing end to end. We used six unlabeled points as the nodes. See Figure 3-1(a). In addition a level set, colored black, is also plotted. It is the estimated level set for $P(y = 1|x) = 0.5$ learned from ℓ_2-regularized logistic regression. A second level set, colored red, also drawn at probability estimate 0.5, is learned from the simultaneous process, with failure cost modeled according to Cost 1.



(a)

(b)

Figure 3-1: Left: x^1 and x^2 represent the first and second coordinates respectively of the 2D feature space. The triangles represent the unlabeled data $\{\tilde{x}_i\}_{i=1}^6$. Right: The numbers in the nodes indicate their probability of failure, and the numbers on the edges indicate distances. The optimal route 1-2-3-6-4-5-1 as determined by the sequential formulation is highlighted.

Now, node 6 (triangle with label "$\tilde{x}_6$") lies in a low density region of feature space, so its probability cannot be well estimated. For the sequential formulation, node 6 was assigned $p(\tilde{x}_6) = 0.5$ and the optimal route obtained by solving the weighted TRP problem is 1-2-3-6-4-5-1, shown in Figure 3-1(b). The node represented by $\tilde{x}_1$ is chosen to be the starting point. For the simultaneous process, node 6 has been assigned a new probability value $p(\tilde{x}_6) = 0.29$. This change is possible because node 6's probability estimate can vary quite a lot without changing the probability estimates of others. This changes the route to 1-2-3-4-5-6-1 as shown in Figure 3-2.

In the simultaneous process, we chose C_1 large enough so that the tour route visits 4 and 5 before 6. This results in a $\sim$ 9% decrease in the failure cost (Cost 1), with a $\sim$ 3% change in the training error (logistic loss). In particular, for the sequential process, Cost 1 is 4.7 units and the training error is 15.7 units; for the simultaneous process, Cost 1 is 4.25 units and the learning error is 16.2 units ($C_1 = 5 \times 10^{-4}$). This is an illustration of the core of MLOC: both predictive models are good, and a range of operational costs and decisions exist between them.

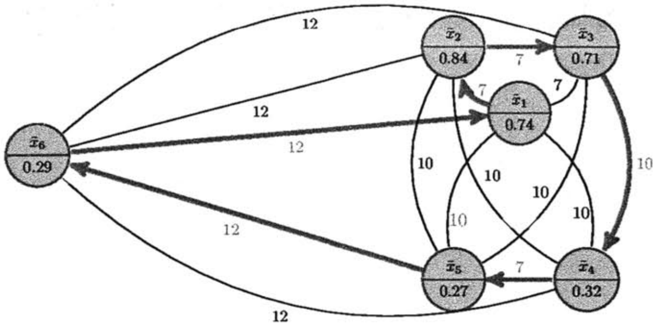


Figure 3-2: The optimal route 1-2-3-4-5-6-1 determined by the simultaneous process is highlighted.

3.5 ML&TRP on the NYC power grid

We now show how the MLOC framework might be used to assist companies like Con Edison, which is NYC's power utility company. We pursue three sets of experiments. The first experiment demonstrates the use of the simultaneous process when given a specific routing problem. This shows how a practitioner would use the simultaneous process in practice. In the second experiment, we randomize over the training sample and routing problems. This experiment shows that the simultaneous process can find models that are equally predictive or better than the sequential method when operational costs are included. In the third experiment, we look at scaling issues.

In all these experiments, we are predicting the probability of failure over the course of a year. While using the predicted failure probabilities in the routing problem, we will assume that these are probabilities of failures in an arbitrary unit interval of time. In particular, they can be the probability of failures over an hour, a day etc. We make the approximation that the probabilities at finer time scales (required for the routing problem) are proportional to the probabilities at coarser time scales for the purpose of our experiments.

3.5.1 The dataset

The dataset we use is described by Rudin et al. [2010], which was developed in order to assist Con Edison with its maintenance and repair programs on the secondary electrical distribution network in NYC; specifically, it was designed for the purpose of predicting manhole fires and explosions. We chose to use all manholes from the Bronx (~23K manholes). Each manhole is represented by (4-dimensional) features that encode the number and type of electrical cables entering the manhole and the number and type of past events involving the manhole. The event features encode how often in the past the manhole was the source of partial outages, full outages and/or underground burnouts. The training features encode events prior to 2008, and the training labels are 1 if the manhole was the source of a serious event (fire, explosion, smoke) during 2008. The prediction task is to predict events in 2009. The test set (for evaluating the performance of the predictive model) consists of features derived from the time period before 2009, and labels from 2009. In our experiments, for both training and test we had a large sample (23,217 instances). There were 211 and 132 failure instances in the test and training data respectively.

3.5.2 Performance of the simultaneous process for a seven node decision problem

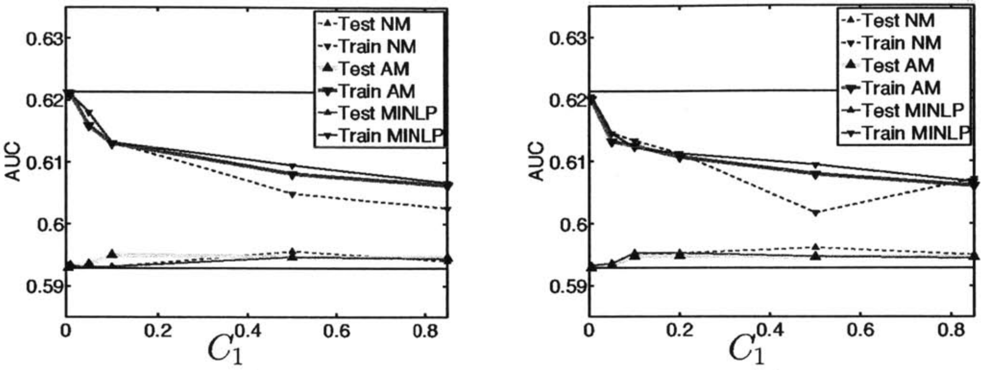
In this experiment, the operational task is to design a route for a repair crew that is equipped to fix seven relatively more vulnerable manholes in 2009. The distances between the nodes were obtained from Google Maps, by querying the driving distance between each pair of nodes. Note that we do not want 'flying' distance between two coordinates as this can be very different from the actual driving distance, especially in New York City.

The limited resources for inspection and repair of manholes should generally be designated to the most vulnerable manholes. With uncertainty in many of the probability estimates, if we are not careful, it is possible that most of these resources will be spent in dealing with outliers whose probabilities are overestimated. The simultaneous process will generally prevent this from happening if we choose C_1 to have a sufficiently large positive value.

Manhole failures are rare events. This means there are many more negative labels than positive labels. Using a logistic model gives probability estimates which are low overall, so the misclassification error is almost always the size of the whole positive class. Because of this, we evaluate the quality of the predictions from f_{λ^*} using the area under the ROC curve (AUC), for both training and test. AUC is a measure of ranking quality; it is sensitive to the rank-ordering of the nodes in terms of their probability to fail, and it is not as sensitive to changes in the values of these probabilities. This means that as the parameter C_1 increases, the estimated probability values will tend to decrease, and thus the failure cost will decrease.

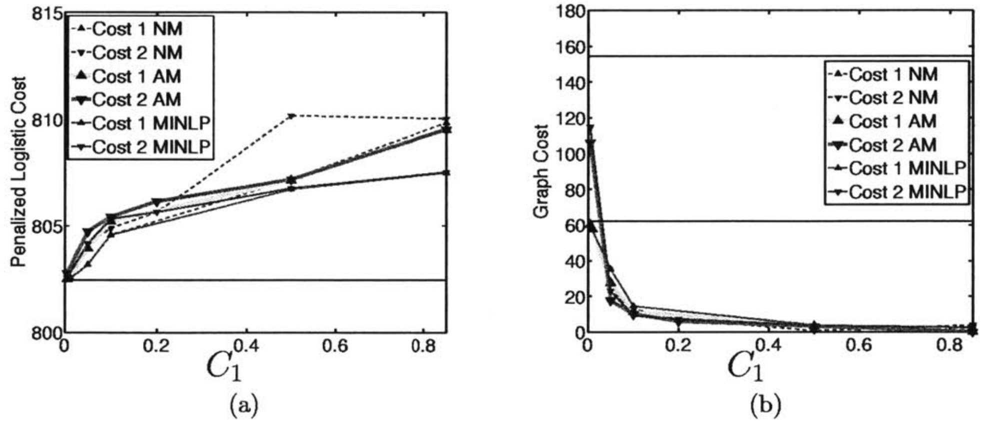
For the experiment, a specific decision problem was sampled and fixed a priori, involving repairs on a handful of relatively more vulnerable manholes in the Bronx. We solved (3.20) and (3.21) for a range of values for the regularization parameter C_1, for both costs and all three methods, with the goal of seeing whether for the same level of estimation performance, we can get a range in the cost of failures. In particular, we wanted to know if we could see a substantial reduction in the cost. We varied C_1 so that the variation in the training error term across the methods was small, about 2% away from the solution of the sequential process ($C_1 = 0$), see Figure 3-4(a). For that range, the test AUC values for the simultaneous process were all within 1% of each other; this is true for both Cost 1 and Cost 2, for each of the AM, NM, and MINLP solvers, see Figures 3-3(a) and 3-3(b). So, changing C_1 did not dramatically impact the prediction quality as measured by the AUC. On the other hand, the failure costs varied widely over the different methods and settings of C_1, as a result of the change in the probability estimates, as shown in Figure 3-4(b). As C_1 was increased from 0.05 to 0.5, Cost 1 went from 27.5 units to 3.2 units, which is over eight times smaller. This means that with a 1-2% variation in the predictive model's AUC, the operational cost can decrease a lot, yielding a completely different possible route for inspection and/or repair work. The reason for an order of magnitude change in the failure cost is because the probability estimates vary by an order of magnitude due to uncertainty at the nodes. This uncertainty in costs is what the MLOC allows us to uncover.

In Figures 3-5(a)-3-5(c) we show the routes according to the different algorithms.



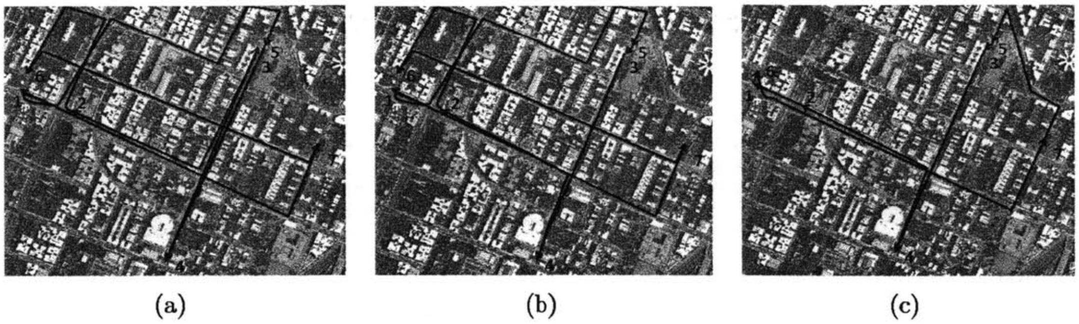
(a) (b)

Figure 3-3: Left: The AUC values corresponding to models (parameterized by C_1) obtained from the simultaneous process using Cost 1 by NM and AM and MINLP techniques. The AUC values on the training data decrease slightly and the same values for test data increase marginally. The two horizontal lines represent the training and test AUC values obtained by ℓ_2-penalized logistic regression are constant with respect to C_1. Right: Similar AUC values obtained from the simultaneous process, using Cost 2.



(a) (b)

Figure 3-4: Left: The ℓ_2-regularized logistic loss increases as a function of increasing C_1. The horizontal line represents the loss value from ℓ_2-penalized logistic regression with no regularization ($C_1 = 0$). Right: The failure costs decrease as a function of the regularization parameter C_1. The horizontal lines in the figure represent the sequential formulation solution; the lower horizontal line is Cost 1 of the solution obtained by ℓ_2-penalized logistic regression, and the upper line is Cost 2 of that solution.



(a) (b) (c)

Figure 3-5: Left: A naïve route: 1-5-4-3-2-6-7-1 obtained by sorting the probability estimates in decreasing order and visiting the corresponding nodes. Center: Sequential process route: 1-5-3-4-2-6-7-1. The simultaneous process also chooses this route when C_1 is small. Right: Route chosen by the simultaneous process when C_1 is larger: 1-6-7-5-3-4-2-1. Prediction performance is only slightly influenced by the route change, but the routing cost (Cost 1) decreases a lot.

We first provide the naïve route in Figure 3-5(a), which was obtained by estimating probabilities using ℓ_2-penalized logistic regression, and then simply visiting nodes according to decreasing values of these probabilities. Figure 3-5(b) shows the route provided by the sequential process. When the failure term starts influencing the optimal solution of the objective (3.20) because of an increase in C_1, we get a new route, depicted in Figure 3-5(c). In most applications relevant to this problem, we suspect that the solution used in practice is somewhere in between the naïve route and the sequential route, in that a human views the naïve solution and adjusts it by hand to be closer to the sequential route (without solving the TRP). For the application to electrical grid maintenance, the simultaneous process was able to find a substantially lower cost route than the naïve or sequential process, with little (if any) change in the AUC prediction quality. This demonstration on data from the Bronx indicates that it is possible to better understand uncertainty in modeling. If engineers truly believe the costs will be lower, their belief, combined with the route we found, can be used to justify a much more cost-effective solution.

3.5.3 Performance of the simultaneous process across randomly generated decision problems

In this experiment, we varied the size of the training data and characterized its effect on learning for both the sequential process and the simultaneous process. We expect to see that when the sample size is small, the operational cost regularization can lead to better performance for the simultaneous process for some C_1. That is, we are showing that some type of knowledge on the operational cost can be helpful in prediction. (When the sample size is large, the regularization term of the simultaneous process should not have much of an effect, and the sequential and simultaneous process models should perform similarly, which is unsurprisingly what we observe.)

To conduct the experiment, we considered training samples ranging from 10% of the original training set size to 100% of the original training set size. For each training set we generated, we then generated 100 seven node decision problems (TRP problems) from a separate held out test set. Each decision problem was generated by randomly picking the nodes (whose labels are not known during training) and computing the distances between each pair of them. For each new training sample size and for each random decision problem, we solved the sequential process and the simultaneous process for both Cost 1 and Cost 2. In particular, this involved the following.

- For the sequential process we performed a 5-fold cross validation to pick the coefficient for the ℓ_2 regularization term. Once the optimal regularization constant was chosen, we computed the predicted probabilities of failure and solved the corresponding weighted TRP subproblem.

- We solved the simultaneous process using the AM algorithm for 4 different C_1 values, and the one achieving the best test performance (on a separate held out test set) was reported. This encodes the notion that one of the C_1 values, namely the one which gives the best test performance, encodes the right prior knowledge. In total, 8,000 mixed integer nonlinear programs were solved (4 C_1 value settings per decision problem (100) per training sample size (10) per decision cost type (Cost 1 and Cost 2)).

Figure 3-6 shows how the simultaneous process compares with respect to the sequential process in terms of AUC on a held out test set as the size of the training sample is varied for Cost 1. The x-axis shows different training sample sizes and the y-axis shows the difference between the AUC of a simultaneous process model (one for each training size and decision problem) and the AUC of the corresponding sequential process model, where 0 means that the AUC's for the two processes were identical. From the figures, we can infer the following:

- The test performance of the simultaneous process can often be better than that of the sequential process for smaller training sets. This is because at lower sample sizes, the simultaneous process gains an advantage from the prior knowledge about operational costs.

- At larger training set sizes, the logistic models from the simultaneous process and the sequential process performed similarly. Again this is not surprising, as the regularization becomes less influential as the training set size increases.

At each training sample size, we tested two hypotheses using the (nonparametric) sign test, with significance level $\alpha = 0.05$. In the first test, the null hypothesis was that the median AUC performance of the two processes was the same versus the alternative that the median AUC performance of the simultaneous process is greater than the median AUC performance of the sequential process. For three of the larger training sample sizes (namely $.6, .7$ and $.9$ of the original), we could not reject the null as the corresponding p-values were greater than the significance level and for the remaining 7 training sample sizes, we could reject the null that the median performance of the two methods is the same. In the second test, the null hypothesis was that the median routing cost using the two processes was the same versus the alternative that the median routing cost of the simultaneous process is smaller than the median routing cost of the sequential process. Here, we were able to reject the null hypothesis for all 10 training sample sizes.

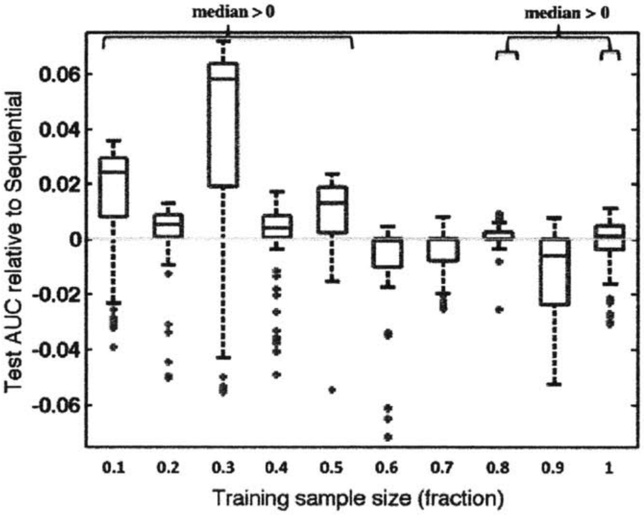


Figure 3-6: Performance of the two processes on randomly generated decision problems at various training sample sizes with Cost 1 as the routing cost. The evaluation is over a separate held out test set. The green solid line is the zero mark. For each size of the training sample on the x-axis (varying from 10% to 100% of the original training sample size), we solved the simultaneous process for 100 random seven node decision problems and the performances of the corresponding models relative to the sequential process models are plotted as a box-plot.

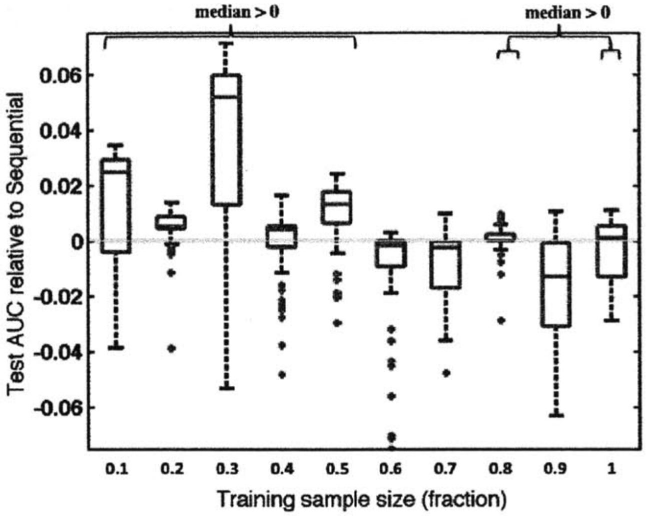


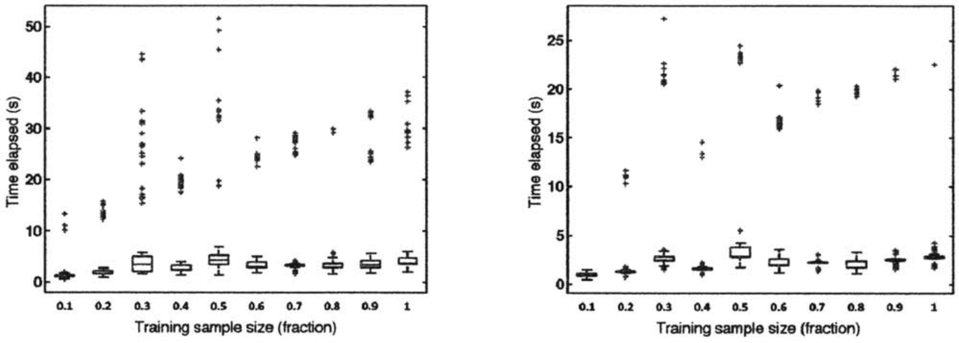
Figure 3-7: Performance of the models output by the two processes on randomly generated decision problems at various training sample sizes with Cost 2 as the routing cost. The evaluation is on a separate held out test set. The green solid line is the zero mark. The box-plots at each training sample size represent the distribution of performances (relative AUC) of the models obtained by the simultaneous process.

We ran this experiment again with Cost 2 as the routing cost, and solved the same 100 decision problems for 4 different C_1 values for each of the 10 different training samples of different sizes. Figure 3.5.2 summarizes the performance of these models. The inferences one can draw from this plot are similar to the previous case.

3.5.4 Scalability of MLOC for Routing

In this experiment, we varied the size of the training sample and decision problem and characterized their effect on time to obtain a solution. All experiments were carried out in a cluster environment (128-256GB RAM, 16-32 core machines).

In the first case, we analyzed the effect of training sample size when the decision problem size was fixed to 7 nodes. In particular, we generated 100 seven node decision problems for each of the 10 training sample sizes (varying from 10% to 100% of the original) and solved the corresponding MINLPs using the AM method discussed in Section 3.4.4. As discussed before, a decision problem was created by randomly



(a)

(b)

Figure 3-8: Left: Boxplot of times taken to solve randomly generated 7 node decision problems for various training sample sizes (from 10% to 100% of the original), when Cost 1 is used. For each training sample size, we solved the simultaneous process for 100 random decision problems and recorded the times. As shown, the time for solving the simultaneous process depends mildly on the size of the training sample size. Right: Boxplot of times taken to solve randomly generated 7 node decision problems for various training sample sizes when Cost 2 is used.

picking a set of seven nodes and computing the distances between them. Additionally, the C_2 parameter was set using 5-fold cross validation. A fixed value of C_1 was also chosen a-priori. Thus a total of 1000 MINLPs were solved for each Cost 1 and Cost 2. Figures 3-8(a) and 3-8(b) show the box plots for the time taken in seconds to solve each simultaneous process problem for Cost 1 and Cost 2 respectively. From the figures, we can infer that as the training sample size increases, the time taken to solve the MINLP increases only mildly for both cost options. This is because the AM method can efficiently scale with the number of examples.

In the second case, we analyzed the effect of decision problem size. In particular, we generated 100 decision problems for node sizes $M = 7, 8, 9, 10, 11, 12, 13$ and 10 decision problems for node size $M = 15$. We solved the MINLPs of Equations (3.20) and (3.21) using the AM method. Similar to the previous experiment, a decision problem of a given size was created by randomly picking a set of nodes and computing the distances between them. The C_2 parameter was set using 5-fold cross validation. The MINLPs were then solved for a fixed value of C_1 chosen a-priori. Thus a total of 710 MINLPs were solved for each Cost 1 and Cost 2. Figures 3-9(a) and 3-9(b) show

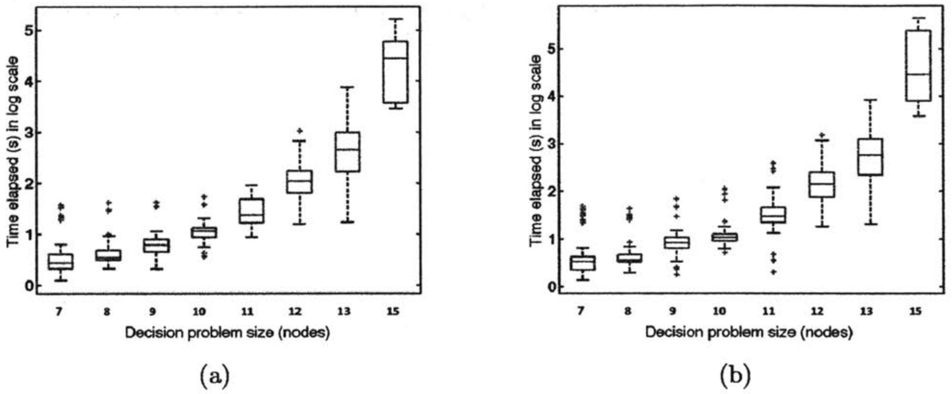


Figure 3-9: Left: Boxplot of times taken to solve the randomly generated decision problems for various values of M, the number of decision problem nodes, when Cost 1 is used. For each decision problem size (varying from 7 to 15), we solved the simultaneous process for 100 random decision problems (10 problems for the 15 node setting) and recorded the times. As shown, the time for solving the decision problem grows exponentially in the size of the decision/routing problem (since the trend is linear in log scale). Right: Boxplot of times taken to solve the randomly generated decision problems for various values of M when Cost 2 is used.

the box plots for the time taken in seconds (in log scale) to solve each simultaneous process problem for Cost 1 and Cost 2 respectively. From the figures, we can infer that as the decision problem size (M nodes) increases, the time taken to solve the MINLP increases exponentially for both cost models. As mentioned earlier, this is because TRP - and generally routing - problems are hard. One needs to solve the TRP anyway, regardless of whether the sequential or simultaneous process is used, to determine the route.

Remark 3.5.1. A note on the performance of other methods (Method 1 and Method 3): For a given C_1, the computation times to solve a typical problem with $\sim 23K$ examples in training and 6, 7, 8, or 10 nodes for the routing problem are about 30, 130, 140, 240 seconds respectively using Method 2 (NM). NM took $\sim$ 1000 iterations to reach a solution where each iteration involved solving a weighted TRP subproblem within $\sim$ 2 seconds. The computation times for solving the MINLP formulation given in (3.20) directly (Method 1) for a given C_1 were $\sim$ 100 times slower. Since the computation times for Method 2 (AM) were the best among the three, we used it to benchmark scalability of MLOC for our application.

3.6 Generalization Bound

We initially introduced the failure cost regularization term in order to find scenarios where the data would support low-cost (more actionable) repair routes. From a learning theoretic point of view, incorporating regularization reduces the size of the hypothesis space and may thus promote generalization. In our case, we can think of decision makers having prior knowledge about how much it should cost for an optimal routing solution. This information should constrain the size of the hypothesis space via the parameter C_1. Increasing C_1 may thus assist in predicting failure probabilities. In what follows, we will provide a generalization bound for the MLOC framework, and specifically for the ML&TRP.

We seek to bound the true risk $R^{\text{true}}(f_\lambda) := E_{(x,y)\sim\mu_{\mathcal{X}\times\mathcal{Y}}} l(f_\lambda(x), y)$ with empirical risk $R^{\text{emp}}(f_\lambda, \{x_i, y_i\}_1^m) = \frac{1}{m}\sum_{i=1}^m l(f_\lambda(x_i), y_i)$ plus a complexity term capturing the size of the hypothesis space. Here $l : f_\lambda(\mathcal{X}) \times \mathcal{Y} \to \mathbb{R}$ is logistic loss, instance (x, y) is drawn from an unknown distribution $\mu_{\mathcal{X}\times\mathcal{Y}}$ and the initial hypothesis space is $\mathcal{F} := \{f_\lambda : f_\lambda(x) = \lambda \cdot x, \lambda \in \mathbb{R}^d, \|\lambda\|_2 \le B_b\}$.

3.6.1 Hypothesis sets for Cost 1 and Cost 2

Consider the ML&TRP with Cost 1 in (3.20). The hypothesis space for the ML&TRP is smaller than $\mathcal{F}$, since we have also the constraint on the failure cost. Replacing the Lagrange multiplier C_1 with an explicit constraint on the failure cost (3.6), we have that for the ML&TRP, f_λ is subject to the failure cost constraint: $\min_\pi \sum_{i=1}^M p(\tilde{x}_{\pi(i)}) L_\pi(\pi(i)) \le C_{\text{budget}}$, where C_{budget} is inversely related to C_1, controlling a “budget" for the failure cost. This gives us the restricted hypothesis space:

$$\mathcal{F}_0 := \left\{ f_\lambda : f_\lambda \in \mathcal{F}, \min_{\pi\in\Pi} \sum_{i=1}^{M} L_\pi(\pi(i)) \frac{1}{1 + e^{-f_\lambda(\tilde{x}_{\pi(i)})}} \le C_{\text{budget}} \right\}.$$

Even though $\mathcal{F}_0$ is smaller than $\mathcal{F}$, it is difficult to construct a tight bound on its covering number. So we enlarge $\mathcal{F}_0$ just enough so that a bound on its covering number can be calculated. In particular, we will enlarge the set $\mathcal{F}_0$ to the set $\mathcal{F}_2$. We define set $\mathcal{F}_2$ parametrized by a vector $a_{\text{budget}} \in \mathbb{R}^d$ as follows:

$$\mathcal{F}_2 := \{f_\lambda : f_\lambda \in \mathcal{F}, a_{\text{budget}} \cdot \lambda \leq 1\},$$

where vector a_{budget} is a function of C_{budget}, the graph and the unlabeled data $\{\tilde{x}_i\}_i$.

$\mathcal{F}_2$ is the intersection of the ball $\mathcal{F}$ with the halfspace defined by a_{budget}; it is a ball that is missing a spherical cap. The vector a_{budget} will capture the effect of C_{budget} in such a way that $\mathcal{F}_0 \subset \mathcal{F}_2$, which we will show within the proof of the Theorem 3.6.2. $\mathcal{F}_2$ is the space whose complexity we will bound, again within the proof of Theorem 3.6.2.

We will now define the vector a_{budget} in terms of C_{budget} and provide a proof later. Let d_i be the shortest distance from the starting node (node 1) to node i for $i = 2, .., M$ and d_1 be the length of the shortest tour that visits all the nodes and returns to node 1. This means $d_i \leq L_\pi(i); i = 1, ..., M$ with equality if the physical graph can be embedded into 1-dimensional Euclidean space. The vector a_{budget} is then related to C_{budget} defined elementwise as:

$$a^j_{\text{budget}} = \frac{1}{C_{\text{budget}} - a_0} \left(\frac{e^{B_b X_b}}{(1 + e^{B_b X_b})^2} \right) \left(\sum_i d_i \tilde{x}^j_i \right) \text{ for } j = 1, .., d \tag{3.22}$$

$$\text{where } a_0 = \left(B_b X_b \frac{e^{B_b X_b}}{(1 + e^{B_b X_b})^2} + \frac{1}{1 + e^{B_b X_b}} \right) \sum_i d_i.$$

Remark 3.6.1. (Defining $\mathcal{F}_0, \mathcal{F}_2$ and a_{budget} for Cost 2): The definitions of $\mathcal{F}_0$ and $\mathcal{F}_2$ can be easily adapted to Cost 2 in (3.21) of the ML&TRP. Here too, the hypothesis space for the ML&TRP is smaller than $\mathcal{F}$ because of the constraint on the failure cost. Again replacing the Lagrange multiplier C_1 with an explicit constraint on the failure cost, we have that for the ML&TRP, f_λ is subject to the failure cost constraint: $\min_\pi \sum_{i=1}^M \log(1 + e^{\lambda \cdot \tilde{x}_{\pi(i)}}) L_\pi(\pi(i)) \leq C_{\text{budget}}$, where C_{budget} is inversely related to C_1, controlling a "budget" for the failure cost. This gives us the restricted hypothesis space:

$$\mathcal{F}_0 := \{f_\lambda : f_\lambda \in \mathcal{F}, \min_{\pi\in\Pi} \sum_{i=1}^{M} L_\pi(\pi(i)) \log(1 + e^{f_\lambda(\tilde{x}_{\pi(i)})}) \leq C_{\text{budget}}\}.$$

We can again enlarge this class of functions just enough so that a bound on the covering number of $\mathcal{F}_0$ can be calculated. The enlarged set $\mathcal{F}_2$ will have the same form as for Cost 1 except for a different definition of a_{budget} (we will derive this later):

$$a^j_{\text{budget}} = \frac{1}{C_{\text{budget}} - a_0} \left(\frac{e^{-B_b X_b}}{1 + e^{-B_b X_b}}\right) \left(\sum_i d_i \tilde{x}^j_i\right) \text{ for } j = 1, .., d \qquad (3.23)$$

$$\text{where } a_0 = \left(B_b X_b \frac{e^{-B_b X_b}}{1 + e^{-B_b X_b}} + \log(1 + e^{-B_b X_b})\right) \sum_i d_i.$$

Since Cost 2 can be handled in the same way as Cost 1, we will focus on Cost 1 for the rest of this section.

3.6.2 Main Generalization Result

Recall that we would like to establish that generalization can depend on C_{budget}. The following theorem shows this explicitly. C_{budget} enters the bound through the vector a_{budget}.

Theorem 3.6.2. (Main Result) Let $\mathcal{X} = \{x \in \mathbb{R}^d : \|x\|_2 \leq X_b\}$, $\mathcal{Y} = \{-1, 1\}$. Let $\mathcal{F}_0$ be defined as above with respect to $\{\tilde{x}_i\}_{i=1}^M$, $\tilde{x}_i \in \mathcal{X}$ (not necessarily random) and a corresponding physical graph. Let $\{x_i, y_i\}_{i=1}^m$ be a sequence of m instances drawn independently according to unknown distribution $\mu_{\mathcal{X}\times\mathcal{Y}}$ and $M_{\text{bound}} := B_b X_b + \log 2$. For any $\epsilon > 0$,

$$P\Big(\exists f \in \mathcal{F}_0 : |R^{\text{emp}}(f_\lambda, \{x_i, y_i\}_1^m) - R^{\text{true}}(f_\lambda)| > \epsilon\Big)$$
$$\leq \; 4\alpha(d, a_{\text{budget}}(C_{\text{budget}})) \left(\frac{32 B_b X_b}{\epsilon} + 1\right)^d \exp\left(\frac{-m\epsilon^2}{128 M^2_{\text{bound}}}\right),$$

where $\alpha(d, a_{\text{budget}}(C_{\text{budget}}))$ is equal to

$$\frac{1}{2}+\frac{\|a_{\text{budget}}\|_2^{-1}+\frac{\epsilon}{32X_b}}{B_b+\frac{\epsilon}{32X_b}}\frac{\Gamma\left[1+\frac{d}{2}\right]}{\sqrt{\pi}\Gamma\left[\frac{d+1}{2}\right]}{}_2F_1\left(\frac{1}{2},\frac{1-d}{2};\frac{3}{2};\left(\frac{\|a_{\text{budget}}\|_2^{-1}+\frac{\epsilon}{32X_b}}{B_b+\frac{\epsilon}{32X_b}}\right)^2\right) \tag{3.24}$$

$$\text{or equivalently, } 1-\frac{1}{2}I_{1-\left(\|a_{\text{budget}}\|_2^{-1}+\frac{\epsilon}{32X_b}\right)^2/\left(B_b+\frac{\epsilon}{32X_b}\right)^2}\left(\frac{d+1}{2},\frac{1}{2}\right) \tag{3.25}$$

and where ${}_2F_1(a,b;c;d)$ and $I_x(a,b)$ are the hypergeometric function and the regularized incomplete beta functions respectively.

The term $\alpha(d, a_{\text{budget}}(C_{\text{budget}}))$ comes directly from formulae for the volume of spherical caps. As C_{budget} decreases, the norm $\|a_{\text{budget}}\|_2$ increases, and thus $\|a_{\text{budget}}\|_2^{-1}$ decreases, (3.24) and (3.25) decrease, and the whole bound decreases. This is the mechanism by which decreasing C_{budget} may improve generalization ability.

Theorem 3.6.2 is specific to the ML&TRP because $\mathcal{F}_0$ was defined based on the ML&TRP and a_{budget} was defined in (3.22) for Cost 1 and (3.23) for Cost 2.

The technique of Theorem 3.6.2 applies much more broadly than the ML&TRP. In fact, we can derive a general bound that applies to any problem with a similar hypothesis space constraint. Specifically, the hypothesis space should be bounded by the intersection of a ball with a half-space.

Corollary 3.6.3. (Bound for General MLOC Framework) Consider any operational cost constraint such that the hypothesis space lies within $\mathcal{F}_2$ defined by $\mathcal{F}_2 = \{f_\lambda \in \mathcal{F} : a_{\text{budget}} \cdot \lambda \leq 1\}$ for some $a_{\text{budget}} \in \mathbb{R}^d$. Then, for any $\epsilon > 0$,

$$\begin{aligned} &P\left(\exists f \in \mathcal{F}_2 : |R^{\text{emp}}(f_\lambda, \{x_i, y_i\}_1^m) - R^{\text{true}}(f_\lambda)| > \epsilon\right) \\ &\leq\ 4\alpha(d, a_{\text{budget}})\left(\frac{32B_bX_b}{\epsilon}+1\right)^d \exp\left(\frac{-m\epsilon^2}{128M_{\text{bound}}^2}\right), \end{aligned}$$

where $\alpha(d, a_{\text{budget}})$ equals

$$\frac{1}{2}+\frac{\|a_{\text{budget}}\|_2^{-1}+\frac{\epsilon}{32X_b}}{B_b+\frac{\epsilon}{32X_b}}\frac{\Gamma\left[1+\frac{d}{2}\right]}{\sqrt{\pi}\Gamma\left[\frac{d+1}{2}\right]}{}_2F_1\left(\frac{1}{2},\frac{1-d}{2};\frac{3}{2};\left(\frac{\|a_{\text{budget}}\|_2^{-1}+\frac{\epsilon}{32X_b}}{B_b+\frac{\epsilon}{32X_b}}\right)^2\right)$$

$$\text{or equivalently, } 1-\frac{1}{2}I_{1-\left(\|a_{\text{budget}}\|_2^{-1}+\frac{\epsilon}{32X_b}\right)^2/\left(B_b+\frac{\epsilon}{32X_b}\right)^2}\left(\frac{d+1}{2},\frac{1}{2}\right)$$

and where ${}_2F_1(a,b;c;d)$ and $I_x(a,b)$ are the hypergeometric function and the regularized incomplete beta functions respectively.

The $\alpha(d, a_{\textbf{budget}})$ is influenced by our belief on the operational cost. Thus, by being able to specify something about the operational cost, we are able to have a better guarantee on generalization. In the case where we are not able to specify anything about the operational cost, the quantity $\alpha(d, a_{\textbf{budget}})$ is equal to 1 giving us the standard generalization result for norm constrained linear function classes.

3.6.3 Proof

The proof outline is as follows. We will construct two classes, $\mathcal{F}_1$ and $\mathcal{F}_2$ that are slightly larger than $\mathcal{F}_0$, but smaller than $\mathcal{F}$ when $C_{\textbf{budget}}$ is small enough. Then we will use a volumetric argument to bound the covering number of $\mathcal{F}_2$, which uses the volumes of spherical caps; the idea is to show that the value of $C_{\textbf{budget}}$ affects the volume of the hypothesis space, and thus the covering number. The covering number bound is then applied to a uniform bound of Pollard [1984] to obtain a generalization bound. The fact that the covering number of $\mathcal{F}_2$ can be below that of $\mathcal{F}$ indicates that using functions from $\mathcal{F}_2$ may provide improvements in generalization over using the full set $\mathcal{F}$.

Let us lead up to the proof of Theorem 3.6.2.

Definition 4. Let $A \subseteq X$ be an arbitrary set and $(X, \mu_{\mathcal{X}\times\mathcal{Y}})$ a (pseudo) metric space. Let $|\cdot|$ denote set size.

- For any $\epsilon > 0$, an ϵ-cover for A is a finite set $U \subseteq X$ (not necessarily $\subseteq A$) s.t. $\forall x \in A, \exists u \in U$ with $\mu_{\mathcal{X}\times\mathcal{Y}}(x,u) \leq \epsilon$.

- A is totally bounded if A has a finite ϵ-cover for all $\epsilon > 0$. The *covering number* of A is $N(\epsilon, A, \mu_{\mathcal{X}\times\mathcal{Y}}) := \inf_{U \in \mathcal{U}} |U|$ where $\mathcal{U}$ is the set of all ϵ-covers for A.

- A set $R \subseteq X$ is ϵ-separated if $\forall x,y \in R, \mu_{\mathcal{X}\times\mathcal{Y}}(x,y) > \epsilon$. The *packing number* $M(\epsilon, A, \mu_{\mathcal{X}\times\mathcal{Y}}) := \sup_{R \in \mathcal{R}} |R|$, where $\mathcal{R}$ is the set of all ϵ-separated subsets of A.

Consider Cost 1. Since, for any collection of values $p(\tilde{x}_i) \geq 0, \sum_i d_i p(\tilde{x}_i) \leq \sum_i L_\pi(i) p(\tilde{x}_i) \leq C_{\text{budget}}$, the class of functions which obey the constraint $\sum_i d_i p(\tilde{x}_i) \leq C_{\text{budget}}$ is larger than the class obeying $\sum_i L_\pi(i) p(\tilde{x}_i) \leq C_{\text{budget}}$. That is, $\mathcal{F}_0 \subseteq \mathcal{F}_1$ where

$$\mathcal{F}_1 := \left\{ f_\lambda : f_\lambda \in \mathcal{F}, \sum_{i=1}^{M} d_i \frac{1}{1+e^{-f_\lambda(\tilde{x}_i)}} \leq C_{\text{budget}} \right\}.$$

As long as $C_{\text{budget}} \leq \sum_{i=1}^{M} d_i$, the constraint in $\mathcal{F}_1$ is not vacuous. The choice of the vector a_{budget} ensures that $\mathcal{F}_1$ is a subset of $\mathcal{F}_2$ as we will prove below.

Lemma 3.6.4. **($\mathcal{F}_0$ is contained in $\mathcal{F}_2$)**

$$N(\epsilon, \mathcal{F}_0, \|\cdot\|_{L_2(\mu_{\mathcal{X}}^m)}) \leq N(\epsilon, \mathcal{F}_1, \|\cdot\|_{L_2(\mu_{\mathcal{X}}^m)}) \leq N(\epsilon, \mathcal{F}_2, \|\cdot\|_{L_2(\mu_{\mathcal{X}}^m)}).$$

Proof. It is sufficient to show $\mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \mathcal{F}_2$. The first inequality was discussed earlier; since $d_i = \inf_{\pi\in\Pi} L_\pi(i)$, this implies:

$$\sum_{i=1}^{M} d_i p(\tilde{x}_i) \leq \sum_{i=1}^{M} L_\pi(i) p(\tilde{x}_i) \leq C_{\text{budget}} \Rightarrow \mathcal{F}_0 \subseteq \mathcal{F}_1.$$

We now show $\mathcal{F}_1 \subseteq \mathcal{F}_2$. We first lower bound $p(\tilde{x}_i)$ by a line with slope $m_1 := \frac{e^{B_b X_b}}{(1+e^{B_b X_b})^2}$ and intercept $m_0 := B_b X_b \frac{e^{B_b X_b}}{(1+e^{B_b X_b})^2} + \frac{1}{1+e^{B_b X_b}}$ such that $m_1 f_\lambda(\tilde{x}_i) + m_0 \leq p(\tilde{x}_i)$ within the function range $[-B_b X_b, B_b X_b]$.

This leads to the definition of a_{budget} as we show now:

$$\sum_i d_i p(\tilde{x}_i) \geq \sum_i d_i (m_1(\lambda \cdot \tilde{x}_i) + m_0) = \tilde{a} \cdot \lambda + a_0, \tag{3.26}$$

$$\text{where} \quad \tilde{a}^j := m_1 \left(\sum_i d_i \tilde{x}_i^j\right) = \frac{e^{B_b X_b}}{(1+e^{B_b X_b})^2} \left(\sum_i d_i \tilde{x}_i^j\right) \text{ for } j = 1, ..., d \tag{3.27}$$

$$\text{and} \quad a_0 = m_0 \sum_i d_i = \left(B_b X_b \frac{e^{B_b X_b}}{(1+e^{B_b X_b})^2} + \frac{1}{1+e^{B_b X_b}}\right) \sum_i d_i.$$

$$\text{Thus} \quad \forall \lambda \in \mathcal{F}_1, \tilde{a} \cdot \lambda + a_0 \leq \sum_{i=1}^{M} d_i p(\tilde{x}_i) \leq C_{\text{budget}}, \tag{3.28}$$

which implies $\tilde{a} \cdot \lambda \leq C_{\text{budget}} - a_0$ or equivalently, $\frac{1}{C_{\text{budget}} - a_0} \tilde{a} \cdot \lambda \leq 1$.

This allows us to define a_{budget} using (3.27) as

$$a^j_{\text{budget}} = \frac{1}{C_{\text{budget}} - a_0}\left(\frac{e^{B_bX_b}}{(1+e^{B_bX_b})^2}\right)\left(\sum_i d_i\tilde{x}^j_i\right) \text{ for } j = 1,..,d,$$

which is the same as (3.22). This vector is such that the set $\mathcal{F}_2$ is larger than $\mathcal{F}_1$. □ □

Remark 3.6.5. (Deriving a_{budget} for Cost 2): The above lemma can be adapted to Cost 2 to give the corresponding a_{budget} that we had defined earlier. In particular, for any collection of values $\log(1+e^{\lambda\cdot\tilde{x}_i}) \geq 0$ for all i,

$$\sum_i d_i \log(1+e^{\lambda\cdot\tilde{x}_i}) \leq \sum_i L_\pi(i)\log(1+e^{\lambda\cdot\tilde{x}_i}).$$

Thus the class of functions that obey the constraint $\sum_i d_i\log(1+e^{\lambda\cdot\tilde{x}_i}) \leq C_{\text{budget}}$ is larger than the class obeying $\sum_i L_\pi(i)\log(1+e^{\lambda\cdot\tilde{x}_i}) \leq C_{\text{budget}}$, which is $\mathcal{F}_0$. $\mathcal{F}_1$ will be the set corresponding to the former constraint:

$$\mathcal{F}_1 := \left\{ f_\lambda \in \mathcal{F} : \sum_{i=1}^{M} d_i \log(1+e^{\lambda\cdot\tilde{x}_i}) \leq C_{\text{budget}} \right\}.$$

We now define $\mathcal{F}_2$ and a_{budget} as follows. We can also see that $\log(1+e^{\lambda\cdot\tilde{x}_i})$ can be lower bounded by a line with slope $m_1 := \frac{e^{-B_bX_b}}{1+e^{-B_bX_b}}$ and intercept $m_0 := B_bX_b\frac{e^{-B_bX_b}}{1+e^{-B_bX_b}} + \log(1+e^{-B_bX_b})$ in the function range $[-B_bX_b, B_bX_b]$ giving us the definition of a_{budget} for Cost 2 as follows:

$$\begin{aligned} C_{\text{budget}} \geq \;& \textstyle\sum_i d_i\log(1+e^{\lambda\cdot\tilde{x}_i}) \geq \sum_i d_i(m_1(\lambda\cdot\tilde{x}_i)+m_0) = \tilde{a}\cdot\lambda + a_0, \\ \text{where} \;\;& \tilde{a}^j := m_1\left(\textstyle\sum_i d_i\tilde{x}^j_i\right) = \tfrac{e^{-B_bX_b}}{1+e^{-B_bX_b}}\left(\textstyle\sum_i d_i\tilde{x}^j_i\right) \text{ for } j = 1, ..., d \\ \text{and} \;\;& a_0 = m_0\textstyle\sum_i d_i = \left(B_bX_b\,\tfrac{e^{-B_bX_b}}{1+e^{-B_bX_b}} + \log(1+e^{-B_bX_b})\right)\textstyle\sum_i d_i. \end{aligned}$$

Thus, $\frac{1}{C_{\text{budget}}-a_0}\tilde{a}\cdot\lambda \leq 1$, and since we wanted to have $a_{\text{budget}}\cdot\lambda \leq 1$ we define a_{budget}

element-wise as:

$$a^j_{\text{budget}} = \frac{1}{C_{\text{budget}} - a_0}\left(\frac{e^{-B_bX_b}}{1+e^{-B_bX_b}}\right)\left(\sum_i d_i \tilde{x}^j_i\right) \text{ for } j = 1,..,d.$$

Note that we have produced two a_{budget} vectors for each of the two costs: Cost 1 and Cost 2 above.

Let $B(0,B_b) := \{\lambda : \lambda \in \mathbf{R}^d, \|\lambda\|_2 \leq B_b\}$. Let the half space corresponding to $\mathcal{F}_2$ be $H_{\|a_{\text{budget}}\|_2^{-1}} := \{\lambda : a_{\text{budget}} \cdot \lambda \leq 1\}$. The lemma below relates covering numbers of $\mathcal{F}$ and $\mathcal{F}_2$ in function space to covering numbers of $B(0,B_b)$ and $B(0,B_b) \cap H_{\|a_{\text{budget}}\|_2^{-1}}$ in $\mathbf{R}^d$.

Lemma 3.6.6. (Relating covering numbers in $\|\cdot\|_{L_2(\mu^m_{\mathcal{X}})}$ to $\|\cdot\|_2$)

a. $\sup_{\mu^m_{\mathcal{X}}} N(\epsilon, \mathcal{F}, \|\cdot\|_{L_2(\mu^m_{\mathcal{X}})}) \leq N(\epsilon/X_b, B(0,B_b), \|\cdot\|_2)$, and

b. $\sup_{\mu^m_{\mathcal{X}}} N(\epsilon, \mathcal{F}_2, \|\cdot\|_{L_2(\mu^m_{\mathcal{X}})}) \leq N(\epsilon/X_b, B(0,B_b) \cap H_{\|a_{\text{budget}}\|_2^{-1}}, \|\cdot\|_2)$.

Proof. Each element $f \in \mathcal{F}$ corresponds to at least one element of $B(0,B_b)$ by definition of $\mathcal{F}$. Choose any distribution $\mu^m_{\mathcal{X}}$. Consider two elements $\lambda_f, \lambda_g \in B(0,B_b)$ corresponding to functions $f, g \in \mathcal{F} \subset L_2(\mu^m_{\mathcal{X}})$. Then,

$$\begin{aligned}
\|f-g\|^2_{L_2(\mu^m_{\mathcal{X}})} &= \frac{1}{m}\sum_{i=1}^m (f(x_i) - g(x_i))^2 \\
&= \frac{1}{m}\sum_{i=1}^m ((\lambda_f - \lambda_g)\cdot x_i)^2 \\
&\leq \frac{1}{m}\sum_{i=1}^m \|\lambda_f - \lambda_g\|_2^2 \|x_i\|_2^2 \text{ (Cauchy-Schwarz to each term)} \\
&\leq \|\lambda_f - \lambda_g\|_2^2 \left(\frac{1}{m}\sum_{i=1}^m X_b^2\right) \text{ (since } \sup_{x\in\mathcal{X}} \|x\|_2 \leq X_b) \\
&= \|\lambda_f - \lambda_g\|_2^2 X_b^2.
\end{aligned}$$

Consider a minimal ϵ/X_b-cover $\{\lambda_r\}_r$ for $B(0,B_b)$ where λ_r corresponds to a function $r \in \mathcal{F}$. Then by definition, $\forall \lambda \in B(0,B_b), \exists \lambda_r : \|\lambda - \lambda_r\|_2 \leq \epsilon/X_b$. Thus, picking any two such elements λ_f, λ_g in a ball of radius ϵ/X_b around λ_r, we see that, the corresponding functions f, g belong to a ball of radius ϵ measured using distance in $L_2(\mu_{\mathcal{X}}^m)$ by the inequality above. The centers of these ϵ-balls in $L_2(\mu_{\mathcal{X}}^m)$ form an ϵ-cover for $\mathcal{F}$. The size of this set is equal to $N(\epsilon/X_b, B(0, B_b), \|\cdot\|_2)$ (which is the size of ϵ/X_b-cover for $B(0, B_b)$). The size of the minimal ϵ-cover of $\mathcal{F}$ will be less than or equal to this size. Hence, $N(\epsilon, \mathcal{F}, \|\cdot\|_{L_2(\mu_{\mathcal{X}}^m)}) \leq N(\epsilon/X_b, B(0, B_b), \|\cdot\|_2)$. Taking a supremum over all $\mu_{\mathcal{X}}^m$, we obtain the first inequality of the lemma. The same argument also works for the second inequality. □ □

Because of rotational symmetry of $B(0, B_b)$, the volume cut off by a hyperplane $a_{\text{budget}} \cdot \lambda = 1$ from $B(0, B_b)$ is determined only by its distance from the origin, which is $1/\|a_{\text{budget}}\|_2$. Such a portion (or its complement, if smaller) of a ball obtained from slicing the ball with a hyperplane is called a spherical cap. It can be parameterized by the distance of its (hyper)plane base from the center of the ball as shown below. For notation, let the volume of a set $A \subset \mathbb{R}^d$ be represented as $Vol(A)$. For example, $Vol(B_1) = \frac{\pi^{d/2}}{\Gamma[d/2+1]}$.

Lemma 3.6.7. (Volume of spherical caps) Let the volume of ball $B(0, B_b)$ in $\mathbb{R}^d$ be denoted as $Vol(B(0, B_b))$. Given a d-dimensional vector a, let $z = \|a\|_2^{-1}$ be a number and $H_z = \{\lambda : a \cdot \lambda \leq 1\}$ be a half space parameterized by z. Let the spherical cap be denoted by $B(0, B_b) \cap H_z'$ where the cap is at a distance z (measured from the base of the cap to the center of the ball), and H_z' represents the complement half space ($H_z \cup H_z' = \mathbb{R}^d$). Then, $Vol(B(0, B_b) \cap H_z')/Vol(B(0, B_b))$ is equal to two expressions:

$$\left(\frac{1}{2} - \frac{z}{B_b} \frac{\Gamma[1+\frac{d}{2}]}{\sqrt{\pi}\Gamma[\frac{d+1}{2}]} {}_2F_1\left(\frac{1}{2}, \frac{1-d}{2}; \frac{3}{2}; \left(\frac{z}{B_b}\right)^2\right)\right) = \frac{1}{2} I_{1-z^2/B_b^2}\left(\frac{d+1}{2}, \frac{1}{2}\right),$$

where ${}_2F_1(a, b; c; d)$ and $I_x(e, f)$ are the hypergeometric and regularized incomplete beta functions respectively.

Proof. See Li [2011] and references therein. □

Next, we need the relationship between packing numbers and covering numbers to prove Theorem 3.6.9:

Lemma 3.6.8. (Packing and covering numbers) For every (pseudo) metric space $(X, \mu_{\mathcal{X}\times\mathcal{Y}})$, $A \subseteq X$, and $\epsilon > 0$,

$$N(\epsilon, A, \mu_{\mathcal{X}\times\mathcal{Y}}) \leq M(\epsilon, A, \mu_{\mathcal{X}\times\mathcal{Y}}).$$

Proof. See Theorem 4 in Kolmogorov and Tikhomirov [1959] or Theorem 12.1 in Anthony and Bartlett [1999] for a proof of this classical result. □

We use the above lemma to obtain bounds for the covering numbers of subsets of $\mathbb{R}^d$ which appeared in Lemma 3.6.6.

Theorem 3.6.9. (Bound on Covering Numbers)

$$\begin{aligned} N(\epsilon/X_b, B(0,B_b), \|\cdot\|_2) &\leq \left(\frac{2B_bX_b}{\epsilon}+1\right)^d, \text{ and} \\ N\left(\epsilon/X_b, B(0,B_b)\cap H_{\|a\|_2^{-1}}, \|\cdot\|_2\right) &\leq \left(\frac{Vol\left(B_{B_b+\frac{\epsilon}{2X_b}} \cap H_{\|a\|_2^{-1}+\frac{\epsilon}{2X_b}}\right)}{Vol\left(B_{B_b+\frac{\epsilon}{2X_b}}\right)}\right)\left(\frac{2B_bX_b}{\epsilon}+1\right)^d. \end{aligned}$$

Proof. Both statements involve a volumetric argument. For a proof of the first inequality, see Section 3 of Kolmogorov and Tikhomirov [1959] or Lemma 4.10 in Pisier [1989] or Lorentz [1966] or Lemma 3 in Cucker and Smale [2002].

To show the second part, let the volume of the complement of the spherical cap be $Vol(B(0,B_b)\cap H_{\|a\|_2^{-1}})$; we need to find an upper bound for the minimal ϵ/X_b-cover of this set. We can do that by scaling a minimal ϵ-cover, which we find now. By extending the boundary of $B(0,B_b)\cap H_{\|a\|_2^{-1}}$ by $\epsilon/2$ we can bound the maximal packing number $M(\epsilon, B(0,B_b)\cap H_{\|a\|_2^{-1}}, \|\cdot\|_2)$ as follows:

$$\begin{aligned} M(\epsilon, B(0,B_b)\cap H_{\|a\|_2^{-1}}, \|\cdot\|_2) &\times Vol(B_1)(\epsilon/2)^d \leq Vol(B_{B_b+\epsilon/2}\cap H_{\|a\|_2^{-1}+\epsilon/2}). \\ \text{Or, } M(\epsilon, B(0,B_b)\cap H_{\|a\|_2^{-1}}, \|\cdot\|_2) &\leq \frac{Vol\left(B_{B_b+\epsilon/2}\cap H_{\|a\|_2^{-1}+\epsilon/2}\right)}{Vol(B_1)}\frac{1}{(\epsilon/2)^d} \\ &= \frac{Vol\left(B_{B_b+\epsilon/2}\cap H_{\|a\|_2^{-1}+\epsilon/2}\right)}{Vol(B_1)}\frac{1}{(\epsilon/2)^d}\frac{(B_b+\epsilon/2)^d}{(B_b+\epsilon/2)^d} \end{aligned}$$

$$= \frac{Vol\left(B_{B_b+\epsilon/2} \cap H_{\|a\|_2^{-1}+\epsilon/2}\right)}{Vol(B_{B_b+\epsilon/2})} \frac{(B_b+\epsilon/2)^d}{(\epsilon/2)^d}.$$

Again, scaling ϵ to ϵ/X_b and using the relationship between $N(\epsilon, A, dist)$ and $M(\epsilon, A, dist)$ in Lemma 3.6.8 yields the second result. □ □

Thus we have so far shown the relationship between covering numbers of $\mathcal{F}_0$, $\mathcal{F}_1$, and $\mathcal{F}_2$ in terms of a certain metric in Lemma 3.6.4, we have shown how those covering numbers are related to covering numbers in $\ell_2(\mathbb{R}^d)$ in Lemma 3.6.6, we have shown how the latter covering numbers relate to volumes in $\ell_2(\mathbb{R}^d)$ in Theorem 3.6.9, and we have shown how to compute one of these volumes in Lemma 3.6.7.

To complete the proof of Theorem 3.6.2, we will use a relation between the covering number of a class of loss functions of some set $\mathcal{G}$ and the covering number of the set $\mathcal{G}$ itself. We will also use a uniform convergence bound of Pollard [1984].

Theorem 3.6.10. (Pollard 1984) Let $l_{\mathcal{G}}$ be a set of functions on $\mathcal{X} \times \mathcal{Y}$ with $0 \leq l(f_\lambda(x), y) \leq M_{\text{bound}}$, $\forall l \in l_{\mathcal{G}}$ and $\forall (x,y) \in \mathcal{X} \times \mathcal{Y}$. Let $\{x_i, y_i\}_1^m$ be a sequence of m instances drawn independently according to $\mu_{\mathcal{X}\times\mathcal{Y}}$. Then for any $\epsilon > 0$,

$$\begin{aligned} &P(\exists l \in l_{\mathcal{G}} : |R^{\text{emp}}(f_\lambda, \{x_i, y_i\}_1^m) - R^{\text{true}}(f_\lambda)| > \epsilon) \\ &\leq \; 4E\left[N\left(\epsilon/16, l_{\mathcal{G}}, \|\cdot\|_{L_1(\mu^m_{\mathcal{X}\times\mathcal{Y}})}\right)\right] \exp\left(\frac{-m\epsilon^2}{128 M^2_{\text{bound}}}\right). \end{aligned}$$

Proof. See Theorem 24 in Pollard [1984] [also in Zhang, 2002, Theorem 1]. □

We can relate the covering number for Pollard's loss functions set $l_{\mathcal{G}}$ to the covering number for set $\mathcal{G}$ as follows.

Lemma 3.6.11. (Relating $l_{\mathcal{G}}$ to $\mathcal{G}$) If every function from function class $l_{\mathcal{G}}$ represented as $l : f(\mathcal{X}) \times \mathcal{Y} \mapsto \mathbb{R}, f \in \mathcal{G}$, is Lipschitz in its first argument with Lipschitz constant $\mathcal{L}$, then the covering number of $l_{\mathcal{G}}$ is related to the covering number of $\mathcal{G}$ by

$$\sup_{\mu^m_{\mathcal{X}\times\mathcal{Y}}} N\left(\epsilon, l_{\mathcal{G}}, \|\cdot\|_{L_1(\mu^m_{\mathcal{X}\times\mathcal{Y}})}\right) \leq N\left(\epsilon/\mathcal{L}, \mathcal{G}, \|\cdot\|_{L_1(\mu^m_{\mathcal{X}})}\right).$$

Proof. Consider two functions $f, g \in \mathcal{G}$. Let the corresponding functions in class $l_{\mathcal{G}}$ be $l_f = l(f(x), y)$ and $l_g = l(g(x), y)$.

$$\begin{aligned} \|l_f - l_g\|_{L_1(\mu^m_{\mathcal{X}\times\mathcal{Y}})} &= \frac{1}{m}\sum_{i=1}^{m} |l(f(x_i), y_i) - l(g(x_i), y_i)| \\ &\leq \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}|f(x_i) - g(x_i)| = \mathcal{L}\|f - g\|_{L_1(\mu^m_{\mathcal{X}})}. \end{aligned}$$

This implies, given $\{x_i, y_i\}_{i=1}^m$, if $\hat{\mathcal{G}}$ is a minimal $\epsilon/\mathcal{L}$-cover of $\mathcal{G}$ in $L_1(\mu^m_{\mathcal{X}})$, we can construct an ϵ-cover of $l_{\mathcal{G}}$ in $L_1(\mu^m_{\mathcal{X}\times\mathcal{Y}})$ as $\hat{l}_{\mathcal{G}} = \{l_{f_i} : f_i \in \hat{\mathcal{G}}\}$. The size of the minimal ϵ-cover will be smaller than the size of such an ϵ-cover. Taking the supremum over all empirical distributions, we get the desired result. □ □

Theorem 3.6.10 and Lemma 3.6.11 involve L_1 covering numbers, but our covering number bounds start with an L_2 metric in Lemma 3.6.6. So we need to switch from L_1 to L_2 metric. The following lemma uses the identity $\|f-g\|_{L_1(\mu^m_{\mathcal{X}})} \leq \|f-g\|_{L_2(\mu^m_{\mathcal{X}})}$ (true because of Jensen's inequality applied to norms) to relate the two.

Lemma 3.6.12. $N(\epsilon, A, \|\cdot\|_{L_1(\mu^m_{\mathcal{X}})}) \leq N(\epsilon, A, \|\cdot\|_{L_2(\mu^m_{\mathcal{X}})})$.

Proof. See for a version, Lemma 10.5 in Anthony and Bartlett [1999]. □

Finally, we can prove the main result.

Proof. (Of Theorem 3.6.2)

In our setting, the loss function is logistic with Lipschitz constant $\mathcal{L} = 1$ (when viewed as a function of $f(x)$). The class of loss functions is thus defined by $l_{\mathcal{F}_0} := \{l : f_\lambda \in \mathcal{F}_0\}$. Each $l \in l_{\mathcal{F}_0}$ is also non-negative and bounded as needed in the statement of Theorem 3.6.10.

Starting from the expectation term on the right hand side of Theorem 3.6.10 using

$\mathcal{F}_0$ as $\mathcal{G}$ we get,

$$\begin{aligned}
&E[N(\epsilon/16, l_{\mathcal{F}_0}, \|\cdot\|_{L_1(\mu^m_{\mathcal{X}\times\mathcal{Y}})})] \\
&\leq \sup_{\mu^m_{\mathcal{X}\times\mathcal{Y}}} N(\epsilon/16, l_{\mathcal{F}_0}, \|\cdot\|_{L_1(\mu^m_{\mathcal{X}\times\mathcal{Y}})}) \text{ bounding expectation by supremum} \\
&\leq \sup_{\mu^m_{\mathcal{X}}} N\left(\frac{\epsilon}{16\mathcal{L}}, \mathcal{F}_2, \|\cdot\|_{L_2(\mu^m_{\mathcal{X}})}\right) \text{ from Lemma 3.6.11, 3.6.12 and 3.6.4 respectively} \\
&\leq N\left(\frac{\epsilon}{16\cdot 1\cdot X_b}, B(0,B_b)\cap H_{\|a_{\text{budget}}\|_2^{-1}}, \|\cdot\|_2\right) \text{ from Lemma 3.6.6, substituting } \mathcal{L}=1 \\
&\leq \left(\frac{Vol\left(B_{B_b+\frac{\epsilon}{32X_b}} \cap H_{\|a_{\text{budget}}\|_2^{-1}+\frac{\epsilon}{32X_b}}\right)}{Vol(B_{B_b+\frac{\epsilon}{32X_b}})}\right)\left(\frac{32B_bX_b}{\epsilon}+1\right)^d \text{ from Theorem 3.6.9} \\
&= \alpha(d, a_{\text{budget}}(C_{\text{budget}}))\left(\frac{32B_bX_b}{\epsilon}+1\right)^d \text{ from Lemma 3.6.7.}
\end{aligned}$$

The above step uses the relation between spherical cap and its complement along with Lemma 3.6.7, $Vol\left(B(0,B_b)\cap H'_{\|a_{\text{budget}}\|_2^{-1}}\right) = Vol(B(0,B_b)) - Vol\left(B(0,B_b)\cap H_{\|a_{\text{budget}}\|_2^{-1}}\right)$.

Using the derived inequality within Theorem 3.6.10 completes the proof. □ □

3.7 Future work

We provide several avenues for future work.

- *Other graph applications:* The MLOC framework is a general tool that can help decision makers translate uncertainty in prediction to uncertainty in operational costs. The ML&TRP itself is a specific application of the MLOC framework that can be applied to the power grid (as we did), but also to delivery truck routing and other physical routing problems, and can be used for more abstract routing problems such as network routing problems, where distances on the graph do not necessarily correspond to a physical distance. In the future it would be interesting to explore some of these applications.

- *Relaxing the cost constraints in the MLOC:* Our generalization bound for the

ML&TRP applied to a hypothesis space that was an intersection of an l_2 ball with a halfspace. It would be interesting to consider more general operational cost constraints, such as quadratic constraints and other convex functions. As it turns out, there are many applications where such constraints naturally arise. In current work, we are constructing bounds for these types of constraints, which lead to exotic hypothesis spaces, such as an intersection of an l_2 ball with an ellipsoid (for quadratic constraints) or a general convex body (for convex constraints).

- *Sequential MLOC:* Currently the MLOC framework applies to one-shot decision problems. It would be interesting to extend it to sequential decision problems, perhaps where multiple decisions are made in a sequence of decision epochs, and training data arrive incrementally. In this case, the baseline technique analogous to the “sequential process" would be a Markov decision process (MDP). The MLOC framework would then assist in understanding the reasonable range of costs for various sequential decision policies. Note that in the current setting, there is no opportunity for exploration to improve our failure estimates. On the other hand, in a sequential MLOC setting, there can be an opportunity to get better failure estimates by collecting information. In such a case, one can take into account the “value of new information” in decision making. Since we do not have a mechanism to collect more information (and update $\{\tilde{x}_i\}_{i=1}^{M}$ and hence, the corresponding failure estimate), we consider only the optimistic and pessimistic decision making approaches in this work.

3.8 Conclusion

In this work, we evaluated the MLOC framework in the context of a real application and demonstrated improvements over current standards. In particular, we presented an application in the domain of transportation routing called the ML&TRP. Our framework takes advantage of uncertainty in statistical modeling to explore the decision space and find potentially more practical solutions. We provide experiments quantifying the improvements and the scalability of the framework with respect to routing problem size. We provided a generalization bound for the ML&TRP (and for the general MLOC framework) indicating that a prior belief in the operational cost can potentially be beneficial to prediction ability in general.

Chapter 4

Generalization Bounds for Learning with Linear, Polygonal, Quadratic and Conic Side Knowledge

4.1 Introduction

Surely, for many applications the amount of domain knowledge we could potentially use within our learning processes is vastly larger than the amount of domain knowledge we actually use. One reason for this is that domain knowledge may be nontrivial to incorporate into algorithms or analysis. A few types of domain knowledge that do permit analysis have been explored quite in depth in the past few years and used very successfully in a variety of learning tasks; this includes knowledge about the sparsity properties of linear models (ℓ_1-norm constraints, minimum description length) or smoothness properties (ℓ_2-norm constraints, maximum entropy). A reason that domain knowledge is not usually incorporated in theoretical analysis is that it can be very problem specific; it may be too specific to the domain to have an overarching theory of interest. For example, researchers in NLP (Natural Language Processing) have long figured out various exotic domain specific knowledge that one can use while performing a learning task Chang et al. [2008a,b]. The present work aims to provide theoretical guarantees for a large class of problems with a general type of domain knowledge that goes beyond sparsity and smoothness.

To define this large class of problems, we will keep the usual supervised learning assumption that the training examples are drawn i.i.d. Additionally in our setting, we have a different set of examples without labels, not necessarily chosen randomly. For this set of unlabeled examples, we have some prior knowledge about the relationships between their labels, which affects the space of hypotheses we are searching over within our learning algorithms. We motivate this knowledge as being obtained from domain experts. These assumptions can, for example, take into account our partial knowledge about how any learned model should predict on the unlabeled examples if they were encountered. We consider many types of side knowledge, namely constraints on the unlabeled examples leading to (i) linear constraints on a linear function class, (ii) quadratic constraints on a linear function class, and (iii) conic constraints on a linear function class. Our main contributions are:

- To show that linear, polygonal, quadratic and conic constraints on a linear hypothesis space can arise naturally in many circumstances, from constraints on a set of unlabeled examples. This is in Section 4.2. We connect these with relevant semi-supervised learning settings.
- To provide upper bounds on covering number and empirical Rademacher complexity for linearly constrained linear function classes. Bounds for the case of linear and polygonal constraints are found in Sections 4.3.3 and 4.3.4 respectively. Two of the three bounds in these sections are not original to this chapter, but their application to general side knowledge with linear constraints is novel.
- To provide two upper bounds on the complexity of the hypothesis space for the quadratic constraint case This can be used directly in generalization bounds. The use of a certain family of circumscribing ellipsoids and the quadratic bounds of Section 4.3.5 are novel to this work.
- To show that one of the upper bounds on the quadratically constrained hypothesis space we provided has a matching lower bound, also in Section 4.3.5. This is novel to this work.

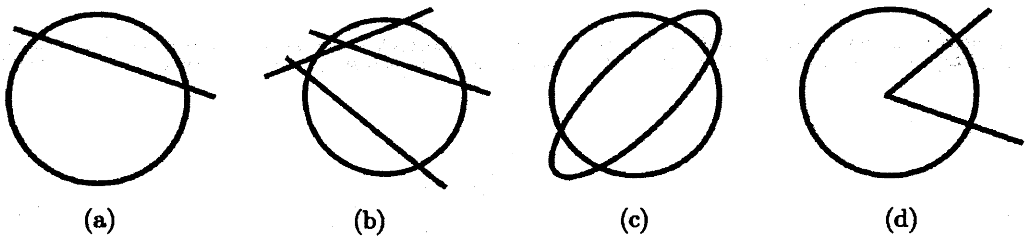


Figure 4-1: This figure illustrates constraints on our hypothesis space. These constraints arise from side knowledge available about a set of unlabeled examples. The ℓ_2 balls in (a), (b), (c) and (d) represent coefficients of linear functions in two dimensions. (a) and (b) represent intersection of a ball and one or several half spaces. Theorems 4.3.1, 4.3.3 and Proposition 4.3.2 analyze these situations. (c) shows the intersection of a ball and an ellipsoid. Theorems 4.3.5, 4.3.7 and 4.3.8 correspond to this setting. (d) shows the intersection of a ball with a second order cone. Theorem 4.3.10 corresponds to this setting.

- To provide a bound on the complexity of the hypothesis space for the conic constraint case. These bounds are in Section 4.3.7 and are novel to this work.
- We develop a novel proof technique for upper bounding linear, quadratic and conic constraint cases based on convex duality.

Figure 4-1 illustrates the various types of side knowledge.

Side knowledge can be particularly helpful in cases where data are scarce; these are precisely circumstances when data themselves cannot fully define the predictive model, and thus domain knowledge can make an impact in predictive accuracy. That said, for any type of side knowledge (sparsity, smoothness, and the side knowledge considered here), the examples and hypothesis space may not conform in reality to the side knowledge. (Similarly, the training data may not be truly random in practice.) However, if they do, we can claim lower sample complexities, and potentially improve our model selection efforts. Thus, we cannot claim that our side knowledge is always true knowledge, but we can claim that if it is true, we are able to gain some benefit in learning.

Motivating examples

Fung et al. [2002] added multiple linear constraints (polygonal constraints) to a specific ERM algorithm, the linear SVM, as a way to incorporate prior knowledge. They investigated the effect of using this type of prior knowledge for classification on a DNA promoter recognition dataset Towell et al. [1990]. In this classification task, the linear constraints result from precomputed rules that are separate from the training data (this is similar to our polygonal setting where constraints are generated from knowledge about the unlabeled examples). The "leave-one-out" error from the 1-norm SVM with the additional constraints was less than that of the plain 1-norm SVM and other training-data-based classifiers such as decision trees and neural networks. This and other types of knowledge incorporation in SVMs are reviewed by Lauer and Bloch [2008] and also Le et al. [2006].

James et al. [2014] motivated the use of linear constraints with LASSO, which is also an ERM procedure. In their experiment, they estimated a demand probability function using an on-line auto lending dataset. They ensured monotonicity of the demand function by applying a set of linear constraints (similar to the poset constraints in 4.2.1) and compared the output to two other methods: logistic regression and the unconstrained LASSO, both of which output non-monotonic demand probability curves.

Nguyen and Caruana [2008a] considered additional unlabeled examples whose labels are partially known. In particular, they worked on a type of multi-class classification task where they know that the label of each unlabeled example belongs to a known subset of the set of all class labels. This knowledge about the unlabeled examples translates into multiple linear constraints (polygonal constraints). They provided experimental results on five datasets showing improvements over multi-class SVMs.

Gómez-Chova et al. [2008] implemented a technique (known as LapSVMs) that uses Laplacian regularization augmented with standard SVMs for two image classification tasks related to urban monitoring and cloud screening (which are both remote sensing tasks). Laplacian regularization means that the regularization term is a quadratic function of the model, derived from a set of unlabeled examples, like our quadratic setting (see Section 4.2.2). In both tasks, the Laplacian-regularized linear SVMs outperformed the standard SVMs in terms of overall accuracy (these improvements are of the order of 2-3% in both cases).

Shivaswamy et al. [2006] formulated robust classification and regression problems as described in Section 4.2.3 leading to conic constraints on the model class. For classification, they used the OCR, Heart, Ionosphere and Sonar datasets from the UCI repository to illustrate the effect of missing values and how robust SVM classification (which introduces second order conic constraints) provides better classification accuracy than the standard SVM classifier after imputation. For regression, they showed improvements in prediction accuracy of a robust version of SVR (again introducing conic constraints on the hypothesis space) as compared to a standard SVR trained after imputation on the Boston housing dataset (also from the UCI repository).

4.2 Linear, Polygonal, Quadratic and Conic Constraints

We are given training sample S of n examples $\{(x_i, y_i)\}_{i=1}^n$ with each observation x_i belong to a set $\mathcal{X}$ in $\mathbb{R}^p$. Let the label y_i belong to a set $\mathcal{Y}$ in $\mathbb{R}$. In addition, we are given a set of m unlabeled examples $\{\tilde{\mathrm{x}}_i\}_{i=1}^m$. We are not given the true labels $\{\tilde{y}_i\}_{i=1}^m$ for these observations. Let $\mathcal{F}$ be the function class (set of hypotheses) of interest, from which we want to choose a function f to predict the label of future unseen observations. Let it be linear, parameterized by coefficient vector β and its description will change based on the constraints we place on β.

Consider the empirical risk minimization problem: $\min_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n l(f(x_i), y_i)$. Here the loss function is a Lipschitz continuous function such as the squared, exponential or hinge loss among others. This supervised learning setup encompasses both supervised classification ($\mathcal{Y}$ is a discrete set) and regression ($\mathcal{Y}$ is equal to $\mathbb{R}$).

Regularization on f acts to enforce assumptions that the true model comes from a restricted class, so that $\mathcal{F}$ is now defined as

$$\{f | f : \mathcal{X} \mapsto \mathcal{Y}, f(x) = \beta^T x, R_l(f) \leq c_l \text{ for } l = 1, ..., L\},$$

where $()^T$ represents the transpose operation. Here we have appended L additional constraints for regularization to the description of the hypothesis set $\mathcal{F}$. Especially if the training set is small, side knowledge can be very powerful in reducing the size of $\mathcal{F}$. Particularly if constants $\{c_l\}_{l=1}^L$ are small, the size of $\mathcal{F}$ be reduced substantially.

4.2.1 Assumptions leading to linear and polygonal constraints

We will provide three settings to demonstrate that linear constraints arise in a variety of natural settings: poset, must-link, and sparsity on $\{\tilde{y}_i\}_{i=1}^m$. In all three, we will include standard regularization of the form $\|\beta\|_q \leq c_1$ by default.

Poset: Partial order information about the labels $\{\tilde{y}_i\}_{i=1}^m$ can be captured via the following constraints: $f(\tilde{\mathbf{x}}_i) \leq f(\tilde{\mathbf{x}}_j) + c_{i,j}$ for any collection of pairs $(i, j) \in [1, ..., m] \times [1, ..., m]$. This gives us up to m^2 constraints of the form $\beta^T(\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j) \leq c_{i,j}$. $\mathcal{F}$ can be described as: $\mathcal{F} := \{f | f(x) = \beta^T x, \|\beta\|_q \leq c_1, \beta^T(\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j) \leq c_{i,j}, \forall (i, j) \in E\}$, where E is the set of pairs of indices of unlabeled data that are constrained.

Must-link: Here we bound the absolute difference of labels between pairs of unlabeled examples: $|f(\tilde{\mathbf{x}}_i) - f(\tilde{\mathbf{x}}_j)| \leq c_{i,j}$. This captures knowledge about the nearness of the labels. This leads to two linear constraints: $-c_{i,j} \leq \beta^T(\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j) \leq c_{i,j}$. These constraints have been used extensively within the semi-supervised Zhu [2005] and constrained clustering settings Lu and Leen [2004], Basu et al. [2006] as must-link or 'in equivalence' constraints. For must-link constraints, $\mathcal{F}$ is defined as: $\mathcal{F} := \{f | f(x) = \beta^T x, \|\beta\|_q \leq c_1, -c_{i,j} \leq \beta^T(\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j) \leq c_{i,j}, \forall (i, j) \in E\}$, where E is again the set of pairs of indices of unlabeled data that are constrained.

Sparsity and its variants on a subset of $\{\tilde{y}_i\}_{i=1}^m$: Similar to sparsity assumptions on β, here we want that only a small set of labels is nonzero among a set of unlabeled examples. In particular, we want to bound the cardinality of the support of the vector $[\tilde{y}_1 \ldots \tilde{y}_{|\mathcal{I}|}]$ for some index set $\mathcal{I} \subset \{1, ..., m\}$. Such a constraint is nonlinear. Nonetheless, a convex constraint of the form $\|[\tilde{y}_1 \ldots \tilde{y}_{|\mathcal{I}|}]\|_1 \leq c_{\mathcal{I}}$ ($2^{|\mathcal{I}|}$ linear constraints) can be used as a proxy to encourage sparsity. The function class is defined as: $\mathcal{F} := \{f | f(x) = \beta^T x, \|\beta\|_q \leq c_1, \|[\beta^T \tilde{\mathbf{x}}_1 \ldots \beta^T \tilde{\mathbf{x}}_{|\mathcal{I}|}]\|_1 \leq c_{\mathcal{I}}\}$. A similar constraint can be obtained if we instead had partial information with respect to the dual norm: $\|[\tilde{y}_1 \ldots \tilde{y}_{|\mathcal{I}|}]\|_\infty \leq c_{\mathcal{I}}$.

4.2.2 Assumptions leading to quadratic constraints

We will provide several settings to show that quadratic constraints arise naturally.

Must-link: A constraint of the form $(f(\tilde{\mathbf{x}}_i) - f(\tilde{\mathbf{x}}_j))^2 \leq c_{i,j}$ can be written as $0 \leq \beta^T A \beta \leq c_{i,j}$ with $A = (\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j)(\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j)^T$. Here A is rank-deficient as it is an outer product, which leads to an unbounded ellipse; however, its intersection with a full ellipsoid (for instance, an ℓ_2-norm ball) is not unbounded and indeed can be a restricted hypothesis set. Set $\mathcal{F}$ is defined by: $\mathcal{F} = \{\beta : \beta^T\beta \leq c_1, \beta^T(\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j)(\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j)^T\beta \leq c_{i,j}; (i,j) \in E\}$, where E is again the set of pairs of indices of unlabeled data that are constrained.

Constraining label values for a pair of examples: We can define the following relationship between the labels of two unlabeled examples using quadratic constraints: if one of them is large in magnitude, the other is necessarily small. This can be encoded using the inequality: $f(\tilde{\mathbf{x}}_i) \cdot f(\tilde{\mathbf{x}}_j) \leq c_{i,j}$. If $f(x) \in \mathcal{Y} \subset \mathbb{R}_+$, then $f(\tilde{\mathbf{x}}_i) \cdot f(\tilde{\mathbf{x}}_j) \leq c_{i,j}$ gives the following quadratic constraint on β with the associated rank 1 matrix being $A = \tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_j^T$: $\beta^T A \beta \leq c_{i,j}$. This is not quite an ellipsoidal constraint yet because matrices associated with ellipsoids are symmetric positive semidefinite. Matrix A on the other hand is not symmetric. Nonetheless, the quadratic constraint remains intact when we replace matrix A with the symmetric matrix $\frac{1}{2}(A + A^T)$. If in addition, the symmetric matrix is also positive-definite (which can be verified easily), then this leads to an ellipsoidal constraint. The hypothesis space $\mathcal{F}$ becomes: $\mathcal{F} = \left\{\beta : \beta^T\beta \leq c_1, \beta^T\tilde{\mathbf{x}}_i\tilde{\mathbf{x}}_j^T\beta \leq c_{i,j}; (i,j) \in E\right\}$.

Energy of estimated labels: We can place an upper bound constraint on the sum of squares (the "energy") of the predictions, which is: $||X_U^T\beta||_2^2 = \sum_i(\beta^T\tilde{\mathbf{x}}_i)^2 = \beta^T(\sum_i \tilde{\mathbf{x}}_i\tilde{\mathbf{x}}_i^T)\beta$ where X_U is a $p \times m$ dimensional matrix with $\tilde{x}_i$'s as its columns.[1] The set $\mathcal{F}$ is $\mathcal{F} = \left\{\beta : \beta^T\beta \leq c_1, ||X_U^T\beta||_2^2 \leq c\right\}$. Extensions like the use of Mahalanobis distance or having the norm act on only a subset of the estimates of $\{\tilde{y}\}_{i=1}^m$ follow accordingly.

Smoothness and other constraints on $\{\tilde{y}_i\}_{i=1}^m$: Consider the general ellipsoid constraint $||\Gamma X_U^T\beta||_2^2 \leq c$ where we have added an additional transformation matrix Γ in front of $X_U^T\beta$. If Γ is set to the identity matrix, we get the energy constraint previously discussed. If Γ is a banded matrix with $\Gamma_{i,i} = 1$ and $\Gamma_{i,i+1} = -1$ for all $i = 1, ..., m$ and remaining entries zero, then we are encoding the side knowledge that the variation in the labels of the unlabeled examples is smoothly varying: we are encouraging the unlabeled examples with neighboring indices to have similar predicted values. This matrix Γ is an instance of a difference operator in the numerical analysis literature. In this context, banded matrices like Γ model discrete derivatives. By including this type of constraint, problems with identifiability and ill-posedness of an optimal solution β are alleviated. That is, as with the Tikhonov regularization on β in least squares regression, constraints derived from matrices like Γ reduce the condition number. The set $\mathcal{F}$ is defined as: $\mathcal{F} = \left\{\beta : \beta^T\beta \leq c_1, ||\Gamma X_U^T\beta||_2^2 \leq c\right\}$.

Graph based methods: Some graph regularization methods such as manifold regularization Belkin and Niyogi [2004] also encode information about the labels of the unlabeled data. They also lead to convex quadratic constraints on β. Here, along with

[1] Note that this notation is not the usual notation where observations $\tilde{x}_i$'s are stacked as rows.

the unlabeled examples $\{\tilde{\mathbf{x}}_i\}_{i=1}^m$, our side knowledge consists of an m-node weighted graph $G = (V, E)$ with the Laplacian matrix $L_G = D - A$. Here, D is a $m \times m$-dimensional diagonal matrix with the diagonal entry for each node equal to the sum of weights of the edges connecting it. Further, A is the adjacency matrix containing the edge weights a_{ij}, where $a_{ij} = 0$ if $(i, j) \notin E$ and $a_{ij} = e^{-c\|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j\|_q}$ if $(i, j) \in E$ (other choices for the weights are also possible). The quadratic function $(X_U^T\beta)^T L_G(X_U^T\beta)$ is then twice the sum over all edges, of the weighted squared difference between the two node labels corresponding to the edge: $2\sum_{(i,j)\in E} a_{ij}\left(f(\tilde{\mathbf{x}}_i) - f(\tilde{\mathbf{x}}_j)\right)^2$. Intuitively, if we have the side knowledge that this quantity is small, it means that a node should have similar labels to its neighbors. For classification, this typically encourages the decision boundary to avoid dense regions of the graph. The set $\mathcal{F}$ is defined as: $\mathcal{F} = \{\beta : \beta^T\beta \le c_1, \beta^T X_U^T L_G X_U^T \beta \le c\}$.

4.2.3 Assumptions leading to conic constraints

We provide two scenarios that naturally lead to conic constraints on the model class: robustness against uncertainty and stochastic constraints.

Robustness against uncertainty in linear constraints: Consider any of the linear constraints considered in Section 4.2.1. All of these can be generically represented as: $\{a_k^T\beta \le 1 \;\; \forall k = 1, .., K\}$ where for each k, a_k is a function of the unlabeled sample $\{\tilde{x}_j\}_{j=1}^m$ (for instance, $a_k = \tilde{x}_i - \tilde{x}_k$ for Poset constraints). Further assume that each a_k is only known to belong to an ellipsoid $\Xi_k = \{\bar{a}_k + A_k u : u^T u \le 1\}$ with both parameters $\bar{a}_k$ and A_k known. This can happen due to measurement limitations, noise and other factors. We want to guarantee that, irrespective of the true value of $a_k \in \Xi_k$, we still have $a_k^T\beta \le 1$.

Borrowing a trick used in the robust linear programming literature, we can encode Lanckriet et al. [2003] the above requirement succinctly as:

$$\bar{a}_k^T\beta + \|A_k^T\beta\|_2 \le 1, \forall k = 1, ..., K$$

which is a set of second-order cone constraints. The feasible set becomes smaller when the linear constraints $\{a_k^T\beta \leq 1 \; \forall k = 1, ..., K\}$ are replaced with the conic constraints above.

Stochastic Programming: Consider a probabilistic constraint of the form $\mathbb{P}_{a_k}(a_k^T\beta \leq 1) \geq \eta_k$, where a_k is now considered a random vector. The motivation for a_k is the same as before (see Section 4.2.1). If we know that a_k is normally distributed (for instance, due to additive noise) with mean $\overline{a}_k$ and covariance matrix B_k, then the probabilistic constraint is the same as: $\overline{a}_k^T\beta + \Phi^{-1}(1-p)\|B_k^{1/2}\beta\|_2 \leq 1$, where $\Phi^{-1}()$ is the inverse error function. To see this, let $u_k = a_k^T\beta$ be a scalar random variable with mean $\overline{u_k}$ and variance σ_k^2 (this is equal to $\beta^T B_k \beta$). Then, our original constraint can be written as $\mathbb{P}\left(\frac{u_k - \overline{u}_k}{\sigma_k} \leq \frac{1-\overline{u}_k}{\sigma_k}\right) \geq \eta_k$. Since $\frac{u_k - \overline{u}_k}{\sigma_k} \sim \mathcal{N}(0,1)$, we can rewrite our constraint as: $\Phi\left(\frac{1-\overline{u}_k}{\sigma_k}\right) \geq \eta_k$ where $\Phi(z)$ is the cumulative distribution function for the standard normal. Further $\Phi\left(\frac{1-\overline{u}_k}{\sigma_k}\right) \geq \eta_k$ implies $\frac{1-\overline{u}_k}{\sigma_k} \geq \Phi^{-1}(\eta_k)$. Rearranging terms, we get $\overline{u}_k + \Phi^{-1}(\eta_k)\sigma_k \leq 1$. Finally, substituting the values for $\overline{u}_k$ and σ_k gives us the following constraint:

$$\overline{a}_k^T\beta + \Phi^{-1}(\eta_k)\|B_k^{1/2}\beta\|_2 \leq 1,$$

which is a second order conic constraint Lobo et al. [1998].

Remark 4.2.1. A question of practical interest would be about ways to impose constraints seen in Sections 4.2.1, 4.2.2 and 4.2.3 in a computationally efficient manner. Fortunately, for all the cases we have considered thus far, the side knowledge can be encoded as a set of convex constraints leading to efficient algorithms (if the original empirical risk minimization problem is convex). Further, note that unlike must-link and similarity side knowledge that lead to convex constraints, cannot-link and dissimilarity knowledge is relatively harder to impose and is typically non-convex.

4.3 Generalization Bounds

In each of the scenarios considered in Section 4.2, essentially we are given m unlabeled examples $\tilde{\mathbf{x}}$ whose subsets satisfy various properties or side knowledge (for instance, linear ordering, quadratic neighborhood similarity, etc). This side knowledge is also shown to constrain the hypothesis space in various ways. In this section, we will attempt to answer the following statistical question: what effect do these constraints have on the generalization ability of the learned model? We will compute bounds on the complexity of the hypothesis space when the types of constraints seen in Section 4.2 are included.

4.3.1 Definition of Complexity Measures

We will look at two complexity measures: the covering number of a hypothesis set and the Rademacher complexity of a hypothesis set. Their definitions are as follows:

Definition 5. *Covering Number Kolmogorov and Tikhomirov [1959]:* Let $A \subseteq \Omega$ be an arbitrary set and (Ω, ρ) a (pseudo-)metric space. Let $|\cdot|$ denote set size. For any $\epsilon > 0$, an **ϵ-cover** for A is a finite set $U \subseteq \Omega$ (not necessarily $\subseteq A$) s.t. $\forall \omega \in A, \exists u \in U$ with $d_\rho(\omega, u) \leq \epsilon$. The **covering number** of A is $N(\epsilon, A, \rho) := \inf_U |U|$ where U is an ϵ-cover for A.

Definition 6. *Rademacher Complexity Bartlett and Mendelson [2002]:* Given a training sample $S = \{x_1, ..., x_n\}$, with each x_i drawn i.i.d. from $\mu_\mathcal{X}$, and hypothesis space $\mathcal{F}$, $\mathcal{F}_{|S}$ is the defined as the restriction of $\mathcal{F}$ with respect to S. The *empirical Rademacher complexity of $\mathcal{F}_{|S}$* is

$$\bar{\mathcal{R}}(\mathcal{F}_{|S}) = \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{F}} \frac{1}{n} \left| \sum_{i=1}^{n} \sigma_i f(x_i) \right| \right]$$

where $\{\sigma_i\}$ are Rademacher random variables ($\sigma_i = 1$ with probability $1/2$ and $\sigma_i = -1$ with probability $1/2$). The *Rademacher complexity of $\mathcal{F}$* is its expectation:

$$\mathcal{R}(\mathcal{F}) = \mathbb{E}_{S \sim (\mu_\mathcal{X})^n} [\bar{\mathcal{R}}(\mathcal{F}_{|S})].$$

If instead we let $\sigma_i \sim \mathcal{N}(0,1)$ in the definition, this is the Gaussian complexity of the function class. Generalization bounds often use both these quantities in their statements Bartlett and Mendelson [2002].

4.3.2 Complexity measures within generalization bounds

Given these definitions, a generalization bound statement can be written as follows Bartlett and Mendelson [2002]: With probability at least $1-\delta$ over the training sample S,

$$\forall f \in \mathcal{F},\ \mathbf{E}_{x,y}[l(f(x),y)] \leq \frac{1}{n}\sum_{i=1}^{n} l(f(x_i),y_i) + 4\mathcal{L}\bar{\mathcal{R}}(\mathcal{F}_{|S}) + O\left(\sqrt{\frac{\log\frac{1}{\delta}}{2n}}\right),$$

where $\mathcal{L}$ is the Lipschitz constant of the loss function l. A relation between the empirical Rademacher complexity and covering number can be used to state the above uniform convergence statement in terms of the covering number. The relation (also known as Dudley's entropy integral) is Talagrand [2005]:

$$\bar{\mathcal{R}}(\mathcal{F}_{|S}) \leq c\int_0^\infty \sqrt{\frac{\log N(\sqrt{n}\epsilon, \mathcal{F}_{|S}, \|\cdot\|_2)}{n}}d\epsilon,$$

where $\mathcal{F}_{|S} = \{(f(x_1),\ldots,f(x_n)) : f \in \mathcal{F}\}$ and c is a constant. Thus, we study upper bounds for covering numbers and empirical Rademacher complexities interchangeably through the rest of the chapter.

4.3.3 Complexity results with a single linear constraint

We state two results: the first is based on volumetric arguments and bounds the covering number and the second is based on convex duality and bounds the empirical Rademacher complexity. The first is a result from Tulabandhula and Rudin [2014] while the second is new to this work.

Volumetric upper bound on the covering number: Tulabandhula and Rudin

[2014] analyzed the setting where a bounded linear function class is further constrained by a half space. The motivation there was to study a specific type of side knowledge, namely knowledge about the cost to solve a decision problem associated with the learning problem. The result there extends well beyond operational costs and is applicable to our setting where we have a ℓ_2 bounded linear function class with a single half space constraint.

Theorem 4.3.1. [Theorem 2 of Tulabandhula and Rudin, 2014] Let $\mathcal{X} = \{x \in \mathbb{R}^p : \|x\|_2 \leq X_b\}$ and $\mu_{\mathcal{X}}$ be the marginal probability measure on $\mathcal{X}$. Let

$$\mathcal{F} = \left\{ f | f : \mathcal{X} \mapsto \mathcal{Y}, f(x) = \beta^T x, \|\beta\|_2 \leq B_b, a^T \beta \leq 1 \right\}.$$

Let $\mathcal{F}_{|S} = \{(f(x_1), \dots, f(x_n)) : f \in \mathcal{F}\}$. Then for all $\epsilon > 0$, for any sample S,

$$N(\sqrt{n}\epsilon, \mathcal{F}_{|S}, \|\cdot\|_2) \leq \alpha(p, a, \epsilon) \left(\frac{2B_b X_b}{\epsilon} + 1 \right)^p.$$

Also, defining $r = B_b + \frac{\epsilon}{2X_b}$ and $V_p(r) = \frac{\pi^{p/2}}{\Gamma(p/2+1)} r^p$, the function α above is:

$$\alpha(p, a, \epsilon) =$$
$$1 - \frac{1}{V_p(r)} \int_{\theta=\cos^{-1}\left(\frac{\|a\|_2^{-1} + \frac{\epsilon}{2X_b}}{r}\right)}^{0} V_{p-1}(r \sin\theta) d(r \cos\theta).$$

Intuition: The function $\alpha(p, a, \epsilon)$ can be considered to be the normalized volume of the ball (which is 1) minus the portion that is the spherical cap cut off by the linear constraint. It comes directly from formulae for the volume of spherical caps. We are integrating over the volume of a $p-1$ dimensional sphere of radius $r \sin\theta$ and the height term is $d(r \cos\theta)$.

This bound shows that the covering number bound can depend on a, which is a direct function of the unlabeled examples $\{\tilde{\mathbf{x}}_i\}_{i=1}^m$. As the norm $\|a\|_2$ increases, $\|a\|_2^{-1}$ decreases, thus $\alpha(p, a, \epsilon)$ decreases, and the whole bound decreases. This is a mechanism by which side information on the labels of the unlabeled examples influences the complexity measure of the hypothesis set, potentially improving generalization.

Relation to standard results: It is known Kolmogorov and Tikhomirov [1959] that $\mathcal{B} = \{\beta : \|\beta\|_2 \leq B_b\}$ has a bound on its covering number of the form $\mathcal{N}(\epsilon, \mathcal{B}, \|\cdot\|_2) \leq \left(\frac{2B_b}{\epsilon} + 1\right)^p$. Since in Theorem 4.3.1 the same term appears, multiplied by a factor that is at most one and that can be substantially less than one, the bound in Theorem 4.3.1 can be tighter.

The above result bounds the covering number complexity for the hypothesis set. Next, we will bound the empirical Rademacher complexity for the same hypothesis set as above.

Convex duality based upper bound on empirical Rademacher complexity: Consider the setting in Theorem 4.3.1. Our attempt to use convex duality to upper bound empirical Rademacher complexity yields the following bound.

Proposition 4.3.2. Let $\mathcal{X} = \{x \in \mathbb{R}^p : \|x\|_2 \leq X_b\}$ and

$$\mathcal{F} = \left\{f | f : \mathcal{X} \mapsto \mathcal{Y}, f(x) = \beta^T x, \|\beta\|_2 \leq B_b, a^T\beta \leq 1\right\}.$$

Then,

$$\bar{\mathcal{R}}(\mathcal{F}_{|S}) \leq \max\left(\mathbb{E}_\sigma\left[\min_{\eta\geq 0}(B_b\|X_L\sigma - \eta a\|_2 + \eta)\right], \mathbb{E}_\sigma\left[\min_{\eta\geq 0}(B_b\|X_L\sigma + \eta a\|_2 + \eta)\right]\right),$$

where $X_L = [x_1 \ \ldots \ x_n]$ is a $p \times n$ dimensional feature matrix and σ is a $n \times 1$ dimensional vector of Bernoulli random variables taking values in $\{-1, 1\}$.

Intuition: We can understand the effect of the linear constraint on the upper bound through the magnitude of vector a. Without loss of generality, let the expectation of the optimal value of the first minimization problem be higher (both minimization problems are structurally similar to each other except for a sign change within the norm term). For a fixed value of σ, this minimization problem involves the distance of vector $X_L\sigma$ to the scaled vector a in the first term and the scaling factor η itself as the second term. Thus, generally, if $\|a\|_2$ is large, the scaling factor η can be small, resulting in a lower optimal value. We also know that larger $\|a\|_2$ corresponds to a tighter half space constraint. Thus, as the linear constraint on the hypothesis space becomes tighter, it makes the optimal solution η and the optimal value smaller for each σ vector. As a result, it tightens the upper bound on the empirical Rademacher complexity.

Relation to standard results: An upper bound on each term of the max operation above can be found by setting $\eta = 0$ that recovers the standard upper bound of $\frac{B_b\sqrt{\text{trace}(X_L^T X_L)}}{\sqrt{n}}$ or $\frac{B_b X_b}{\sqrt{n}}$ without capturing the effect of the linear constraint $a^T\beta \leq 1$.

4.3.4 Complexity results with polygonal/multiple linear constraints and general norm constraints

The following result is from Tulabandhula and Rudin [2013], where the authors analyze the effect of decision making bias on generalization of learning. Again, as in the single linear constraint case, the result extends beyond the setting considered in that paper. In particular, it covers all the motivating scenarios described in Section 4.2.1.

Let us define the matrix $[x_1 \ \dots \ x_n]$ as matrix X_L where $x_i \in \mathcal{X} = \{x : \|x\|_r \leq X_b\}$. Then, X_L^T can be written as $[h_1 \cdots h_p]$ with $h_j \in \mathbb{R}^n, j = 1, ..., p$. Define function class $\mathcal{F}$ as

$$\begin{aligned}\mathcal{F} = \Big\{ & f | f(x) = \beta^T x, \beta \in \mathbb{R}^p, \|\beta\|_q \leq B_b, \\ & \sum_{j=1}^{p} c_{j\nu}\beta_j + \delta_\nu \leq 1, \delta_\nu > 0, \nu = 1, ..., V \Big\},\end{aligned}$$

where $1/r + 1/q = 1$ and $\{c_{j\nu}\}_{j,\nu}$, $\{\delta_\nu\}_\nu$ and B_b are known constants. In other words, we have V linear constraints in addition to a ℓ_q norm constraint. As before, let $\mathcal{F}_{|S}$ be the restriction of $\mathcal{F}$ with respect to S.

Let $\{\tilde{c}_{j\nu}\}_{j,\nu}$ be proportional to $\{c_{j\nu}\}_{j,\nu}$ in the following manner:

$$\tilde{c}_{j\nu} \;:=\; \frac{c_{j\nu}n^{1/r}X_bB_b}{\|h_j\|_r} \quad \forall j = 1, ..., p \text{ and } \nu = 1, ..., V.$$

Let K be a positive number. Further, let the sets P^K parameterized by K and P_c^K parameterized by K and $\{\tilde{c}_{j\nu}\}_{j,\nu}$ be: $P^K := \left\{(k_1, ..., k_p) \in \mathbb{Z}^p : \sum_{j=1}^p |k_j| \leq K\right\}$, and $P_c^K := \left\{(k_1, ..., k_p) \in P^K : \sum_{j=1}^p \tilde{c}_{j\nu}k_j \leq K \; \forall \nu = 1, ..., V\right\}$. Let $|P^K|$ and $|P_c^K|$ be the sizes of the sets P^K and P_c^K respectively. The subscript c in P_c^K denotes that this polyhedron is a constrained version of P^K. Define X_{sL} to be equal to the product of a diagonal matrix (whose j^{th} diagonal element is $\frac{n^{1/r}X_bB_b}{\|h_j\|_r}$) and X_L. Define $\lambda_{\min}(X_{sL}{X_{sL}}^T)$ to be the smallest eigenvalue of the matrix $X_{sL}{X_{sL}}^T$.

Theorem 4.3.3. [Theorem 6 of Tulabandhula and Rudin, 2013]

$$N(\sqrt{n}\epsilon, \mathcal{F}_{|S}, \|\cdot\|_2) \leq \begin{cases} \min\{|P^{K_0}|, |P_c^K|\} & \text{if } \epsilon < X_bB_b \\ 1 & \text{otherwise} \end{cases},$$

where $K_0 = \left\lceil \frac{X_b^2B_b^2}{\epsilon^2} \right\rceil$ and K is the maximum of K_0 and

$$\left\lceil \frac{nX_b^2B_b^2}{\lambda_{\min}(X_{sL}{X_{sL}}^T)\left[\min_{\nu=1,...,V} \frac{\delta_\nu}{\sum_{j=1}^p |\tilde{c}_{j\nu}|}\right]^2} \right\rceil.$$

Intuition: The linear assumptions on the labels of the unlabeled examples $\{\tilde{\mathbf{x}}_i\}_{i=1}^m$ determine the parameters $\{\tilde{c}_{j\nu}\}_{j,\nu}$ that in turn influence the complexity measure bound. In particular, as the linear constraints given by the $c_{j\nu}$'s force the hypothesis space to be smaller, they force $|P_c^K|$ to be smaller. This leads to a tighter upper bound on the covering number.

Relation to standard results: We recover the covering number bound for linear function classes given in Zhang [2002] when there are no linear constraints. In this case, the polytope P^K is well structured and the number of integer points in it can be upper bounded in an explicit way combinatorially.

It is possible to convex duality to upper bound the empirical Rademacher complexity as we did in Proposition 4.3.2. However, the intuition is less clear, and thus, we omit the bound here.

4.3.5 Complexity results with quadratic constraints

Consider the set $\mathcal{F} = \{f : f = \beta^T x, \beta^T A_1 \beta \leq 1, \beta^T A_2 \beta \leq 1\}$. Assume that at least one of the matrices is positive definite and both are positive-semidefinite, symmetric. Let $\Xi_1 = \{\beta : \beta^T A_1 \beta \leq 1\}$ and $\Xi_2 = \{\beta : \beta^T A_2 \beta \leq 1\}$ be the corresponding ellipsoid sets.

Upper bound on empirical Rademacher complexity: We first find an ellipsoid $\Xi_{\text{int}\gamma}$ (with matrix $A_{\text{int}\gamma}$) circumscribing the intersection of the two ellipsoids Ξ_1 and Ξ_2 and then find a bound on the Rademacher complexity of a corresponding function class leading to our result for the quadratic constraint case. We will pick matrix $A_{\text{int}\gamma}$ to have a particularly desirable property, namely that it is *tight*. We will call a circumscribing ellipsoid *tight* when no other ellipsoidal boundary comes between its boundary and the intersection $(\Xi_1 \cap \Xi_2)$. If we thus choose this property as our criterion for picking the ellipsoid, then according to the following result, we can do so by a convex combination of the original ellipsoids:

Theorem 4.3.4. [Circumscribing ellipsoids, Kahan, 1968] There is a family of circumscribing ellipsoids that contains every tight ellipsoid. Every ellipsoid $\Xi_{\text{int}\gamma}$ in this family has $\Xi_{\text{int}\gamma} \supseteq (\Xi_1 \cap \Xi_2)$ and is generated by matrix $A_{\text{int}\gamma} = \gamma A_1 + (1-\gamma) A_2$, $\gamma \in [0, 1]$.

Using the above theorem, we can find a tight ellipsoid $\{\beta : \beta^T A_{\text{int}\gamma} \beta \leq 1\}$ that contains the set $\{\beta : \beta^T A_1 \beta \leq 1, \beta^T A_2 \beta \leq 1\}$ easily. Note that the right hand sides of the quadratic constraints defining these ellipsoids can be equal to one without loss of generality.

Theorem 4.3.5. (Rademacher complexity of linear function class with two quadratic constraints) Let

$$\mathcal{F} = \{f : f(x) = \beta^T x : \beta^T \mathbb{I} \beta \leq B_b^2, \beta^T A_2 \beta \leq 1\}$$

with A_2 symmetric positive-semidefinite. Then,

$$\bar{\mathcal{R}}(\mathcal{F}_{|S}) \leq \frac{1}{n}\sqrt{\text{trace}(X_L^T A_{\text{int}\gamma}^{-1} X_L)}, \tag{4.1}$$

where $A_{\text{int}\gamma}$ is the matrix of a circumscribing ellipsoid $\{\beta : \beta^T A_{\text{int}\gamma} \beta \leq 1\}$ of the set $\{\beta : \beta^T \mathbb{I} \beta \leq B_b^2, \beta^T A_2 \beta \leq 1\}$ and X_L is the matrix $[x_1 \ \ldots \ x_n]$ with examples x_i's as its columns.

Intuition: If the quadratic constraints are such they correspond to small ellipsoids, then the circumscribing ellipsoid will also be small. Correspondingly, the eigenvalues of $A_{\text{int}\gamma}$ will be large. Since, the upper bound depends inversely on the magnitude of these eigenvalues (since it depends on $A_{\text{int}\gamma}^{-1}$), it becomes tighter. Also, in the setting where the original ellipsoids are large and elongated but their intersection region is small and can be bounded by a small circumscribing ellipsoid, the upper bound is again tighter.

Relation to standard results: If $A_{\text{int}\gamma}$ is diagonal (or axis-aligned), then we can write the empirical complexity $\bar{\mathcal{R}}(\mathcal{F}_{|S})$ in terms of the eigenvalues $\{\lambda_i\}_{i=1}^p$ as $\bar{\mathcal{R}}(\mathcal{F}_{|S}) \leq \frac{1}{n}\sqrt{\sum_{j=1}^n \sum_{i=1}^p \frac{x_{ji}^2}{\lambda_i}}$ and this can be bounded by $\frac{X_b B_b}{\sqrt{n}}$ Kakade et al. [2008] when $A_2 = \mathbf{0}$. In that case, all of the λ_i are $\frac{1}{B_b^2}$.

Remark 4.3.6. Since we can choose any circumscribing matrix $A_{\text{int}\gamma}$ in this theorem, we can perform the following optimization to get a circumscribing ellipsoid that minimizes the bound:

$$\min_{\gamma \in [0,1]} \text{trace}(X_L^T (\gamma A_1 + (1-\gamma) A_2)^{-1} X_L). \tag{4.2}$$

This optimization problem is a univariate non-linear program.

Lower bound on empirical Rademacher complexity: We will now show that the dependence of the complexity on $A_{\text{int}\gamma}^{-1}$ is near optimal.

Since $A_{\text{int}\gamma}$ is a real symmetric matrix, let us decompose $A_{\text{int}\gamma}$ into a product $P^T DP$ where D is a diagonal matrix with the eigenvalues of $A_{\text{int}\gamma}$ as its entries and P is an orthogonal matrix (i.e., $P^T P = I$). Our result, which is similar in form to the upper bound of Theorem 4.3.5, is as follows.

Theorem 4.3.7.

$$\bar{\mathcal{R}}(\mathcal{F}_{|S}) \geq \frac{\kappa}{n \log n} \sqrt{\text{trace}(X_L^T A_{\text{int}\gamma}^{-1} X_L)}$$

where

$$\kappa = \frac{1}{C\sqrt{1 + \frac{2\pi p n X_b^2}{(\min_{j=1,\dots,p} \|(PX_L)_j\|_2)^2}}},$$

C is the constant in Lemma 4.5.5, P is the orthogonal matrix from the decomposition of $A_{\text{int}\gamma}$, p, X_b are problem constants and n is the number of training examples.

Intuition: The lower bound is showing that the dependence on $\sqrt{\text{trace}(X_L^T A_{\text{int}\gamma}^{-1} X_L)}$ is tight modulo a $\log n$ factor and a factor (κ). The $\log n$ factor is essentially due to the use of the relation between Gaussian and Rademacher complexities in our proof technique. On the other hand, κ depends on the interaction between the side knowledge about the unlabeled examples (captured through matrix P) and the feature matrix X_L. If there is no interaction, that is, PX_L has zero valued rows for all $j = 1, ..., p$, then the lower bound on empirical Rademacher complexity becomes equal to 0. On the other hand, when there is higher interaction between $A_{\text{int}\gamma}$ (or equivalently, P) and X_L, then the factor κ grows larger, tightening the lower bound on the empirical Rademacher complexity.

The dependence of the lower bound on the strength of the additional convex quadratic constraint is captured via $A_{\text{int}\gamma}$ and behaves in a similar way to the upper bound. That is, when the constraint leads to a small circumscribing ellipsoid, the eigenvalues of $A_{\text{int}\gamma}^{-1}$ are small and the lower bound is small (just like the upper bound).

On the other hand, if the constraint leads to a larger circumscribing ellipsoid, the eigenvalues of $A_{\text{int}\gamma}^{-1}$ are large, leading to a higher values of the lower bound (the upper bound also increases similarly).

Relation to standard results: As with the upper bound, when there is no second quadratic constraint, $A_{\text{int}\gamma} = \frac{1}{B_b^2}\mathbf{I}$. The lower bound depends on the training data through the term $\sqrt{\text{trace}(X_L^T X_L)}$ in this case.

Comparison to the upper bound: For comparison, we see that the upper bound in Theorem 4.3.5 is of the form $\frac{1}{n}\sqrt{\text{trace}(X_L^T A_{\text{int}\gamma}^{-1} X_L)}$ while the lower bound of Theorem 4.3.7 is of the form

$$\frac{\kappa}{n\log n}\sqrt{\text{trace}(X_L^T A_{\text{int}\gamma}^{-1} X_L)},$$

where κ depends on $A_{\text{int}\gamma}$ and X_L.

The proof for the lower bound is similar to what one would do for estimating the complexity of a ellipsoid itself (without regard to a corresponding linear function class). See also the work of Wainwright [2011] for handling single ellipsoids.

Comparison of empirical Rademacher complexity upper bound with a covering number based bound: When matrix $A_{\text{int}\gamma}$ describing a circumscribing ellipsoid has eigenvalues $\{\lambda_i\}_{i=1}^p$, then the covering number can be bounded as:

$$N(\sqrt{n}\epsilon, \mathcal{F}_{|S}, \|\cdot\|_2) \leq \Pi_{i=1}^p \left(\frac{2X_b}{\epsilon\sqrt{\lambda_i}} + 1\right).$$

To get a tight bound, among all circumscribing ellipsoids, we should pick one that minimizes the right hand side of the bound. To do this, we solve an optimization problem involving volume minimization that is different than in (4.2). For instance, this volume minimization can be done using the following steps if at least one of the matrices among A_1 and A_2 is positive-definite:

- First, A_1 and A_2 are simultaneously diagonalized by congruence (say with a non-singular matrix called C) to obtain diagonal matrices $\text{Diag}(a_{1i})$ and $\text{Diag}(a_{2i})$. We can guarantee that the set of ratios $\{\frac{a_{1i}}{a_{2i}}\}$ obtained will be unique.

- The desired ellipsoid $A_{\text{int}\gamma^*}$ can then be obtained by computing

$$\gamma^* \in \arg\max_{\gamma\in[0,1]} \Pi_{i=1}^{p}(\gamma a_{1i} + (1-\gamma)a_{2i})$$

and then multiplying the optimal diagonal matrix $\text{Diag}(\gamma^* a_{1i} + (1-\gamma^*)a_{2i})$ with the congruence matrix C appropriately. Optimal γ^* can be found in polynomial time (for example, using Newton-Raphson).

Comparison with the duality approach to upper bounding empirical Rademacher complexity: A convex duality based upper bound can be derived as shown below.

Theorem 4.3.8. Consider the setting of Theorem 4.3.5. Then,

$$\bar{\mathcal{R}}(\mathcal{F}_{|S}) \le \inf_{\eta\in[0,1]} \left\{ \frac{1}{4n}\text{trace}(X_L^T A_{\text{int}\eta}^{-1} X_L) + \frac{1}{n}(B_b^2 + \eta(1-B_b^2)) \right\}, \tag{4.3}$$

where $A_{\text{int}\eta} = \mathbb{I} + \eta(A_2 - \mathbb{I})$.

This upper bound looks similar to the result in Equation (4.1). Note that $A_{\text{int}\eta}$ is different from $A_{\text{int}\gamma}$ in Theorem 4.3.5. $A_{\text{int}\gamma}$ comes from a circumscribing ellipsoid, whereas $A_{\text{int}\eta}$ does not. Instead, the matrix $A_{\text{int}\eta}$ is picked such that η minimizes the right hand side of the bound in Equation 4.3. Qualitatively, we can see that if the matrix A_2 corresponding to the second ellipsoid constraint has large eigenvalues (for instance, when the second ellipsoid is a smaller sphere, or is an elongated thin ellipsoid), then $A_{\text{int}\eta}^{-1}$ is 'small' (the eigenvalues are small) leading to a tighter upper bound on the empirical Rademacher complexity.

Extension to multiple convex quadratic constraints: Although Section 4.3.5 deals with only two convex quadratic constraints, the same strategy can be used to upper bound the complexity of hypothesis class constrained by multiple convex quadratic constraints. In particular, let $\mathcal{F} = \{f : f = \beta^T x, \beta^T A_k \beta \le 1 \ \ \forall k = 1, ..., K\}$. Again, assume one of the matrices A_k is positive definite. We can approach this problem in two stages. In the first step, we find an ellipsoid $\Xi_{\text{int}\gamma}$ (with matrix $A_{\text{int}\gamma}$) circumscribing the intersections of the K original ellipsoids and in the second step, we reuse Theorem 4.3.5 to obtain an upper bound in $\bar{\mathcal{R}}(\mathcal{F}_{|S})$.

We will generalize Equation (4.2) to look for a circumscribing ellipsoid from the family of ellipsoids parameterized by a K dimensional vector γ constrained to the $K-1$ simplex. In other words, the family of circumscribing ellipsoids is given by $\{\beta^T A_{\text{int}\gamma}\beta \leq 1 : A_{\text{int}\gamma} = \sum_{k=1}^K \gamma_k A_k, \sum_{k=1}^K \gamma_k = 1, \gamma_k \geq 0 \ \ \forall k = 1, ..., K\}$. We can pick one circumscribing ellipsoid from this family by minimizing the right hand side of Equation 4.1 over the $K-1$ simplex similar to Equation (4.2):

$$\min_{\gamma \in \left\{\gamma : \sum_{k=1}^K \gamma_k = 1, \gamma_k \geq 0 \ \forall k=1,...,K\right\}} \text{trace}\left(X_L^T \left(\sum_{k=1}^K \gamma_k A_k\right)^{-1} X_L\right).$$

The above optimization problem is a $K-1$ dimensional polynomial optimization problem.

4.3.6 Complexity results with linear and quadratic constraints

Consider now the setting where we have both linear and quadratic constraints. In particular, we can have the assumptions leading to linear constraints and those leading to quadratic constraints hold simultaneously. In such a setting, based on Theorems 4.3.3 and 4.3.4, we can get a potentially tighter covering number result as follows. Let $x_i \in \mathcal{X} = \{x : \|x\|_2 \leq X_b\}$. Let the function class $\mathcal{F}$ be

$$\mathcal{F} = \Big\{f | f(x) = \beta^T x, \beta \in \mathbb{R}^p, \beta^T A_1 \beta \leq 1, \beta^T A_2 \beta \leq 1,$$
$$\sum_{j=1}^p c_{j\nu}\beta_j + \delta_\nu \leq 1, \delta_\nu > 0, \nu = 1, ..., V\Big\},$$

where $\{c_{j\nu}\}_{j,\nu}$, $\{\delta_\nu\}_\nu$, A_1 and A_2 are known beforehand.

Let matrix $A_{\text{int}\gamma}$ be such that $\{\beta : \beta^T A_1 \beta \leq 1, \beta^T A_2 \beta \leq 1\}$ is circumscribed by $\{\beta : \beta^T A_{\text{int}\gamma}\beta \leq 1\}$. Defining $\{\tilde{c}_{j\nu}\}$ and X_{sL} in the same way as in Section 4.3.3, we get the following corollary.

Corollary 4.3.9. (of Theorem 4.3.3)

$$N(\sqrt{n}\epsilon, \mathcal{F}_{|S}, \|\cdot\|_2) \leq \begin{cases} \min\{|P^{K_0}|, |P_c^K|\} & \text{if } \epsilon < X_b\sqrt{\lambda_{\max}(A_{\text{int}\gamma}^{-1})} \\ 1 & \text{otherwise} \end{cases}.$$

Here, $K_0 = \left\lceil \frac{X_b^2 \lambda_{\max}(A_{\text{int}\gamma}^{-1})}{\epsilon^2} \right\rceil$ and K is the maximum of K_0 and

$$\left\lceil \frac{n X_b^2 \lambda_{\max}(A_{\text{int}\gamma}^{-1})}{\lambda_{\min}(X_{sL}{X_{sL}}^T)\left[\min_{\nu=1,\dots,V} \frac{\delta_\nu}{\sum_{j=1}^p |\tilde{c}_{j\nu}|}\right]^2} \right\rceil.$$

The corollary holds for any $A_{\text{int}\gamma}$ that satisfies the circumscribing requirement. In particular, we can construct the ellipsoid $\{\beta : \beta^T A_{\text{int}\gamma}\beta \leq 1\}$ such that it 'tightly' circumscribes the set $\{\beta : \beta^T A_1 \beta \leq 1, \beta^T A_2 \beta \leq 1\}$ using Theorem 4.3.4 in the same way as we did in Section 4.3.5. The intuition for how the parameters of our side knowledge, namely, the linear inequality coefficients and the matrices corresponding to the ellipsoids, is the same as in Sections 4.3.4 and 4.3.5. Relation to standard results have also been discussed in these sections.

Extension to arbitrary convex constraints: There are at least three ways to reuse the results we have with linear, polygonal, quadratic and conic constraints to give upper bounds on covering number or empirical Rademacher complexity of function classes with arbitrary convex constraints. Such arbitrary convex constraints can arise in many settings. For instance, when the convex quadratic constraints in Section 4.2.2 are not symmetric around the origin, we cannot use the results of Section 4.3.5 directly, but the following techniques apply. Other typical convex constraints include those arising from likelihood models, entropy biases and so on.

The first approach involves constructing an outer polyhedral approximation of the convex constraint set. For instance, if we are given a separation oracle for the convex constraint, constructing an outer polyhedral approximation is relatively straightforward. We can also optimize for properties like the number of facets or vertices of the polyhedron during such a construction. Given such an outer approximation, we can apply Theorem 4.3.3 to get an upper bound on the covering number of the hypothesis space with the given convex constraint.

The second approach involves constructing a circumscribing ellipsoid for the constraint set. This is possible for any convex set in general John [1948]. In addition if the convex set is symmetric around the origin, the 'tightness' of the circumscribing ellipsoid improves by a factor $\sqrt{p}$, where p is the dimension of the linear coefficient vector β. Given such a circumscribing ellipsoid, we can apply Theorem 4.3.5 to get an upper bound on the empirical Rademacher complexity of the original function class with the convex constraint. The quality of both of these outer relaxation approaches depends on the structure and form of the convex constraint we are given.

The third approach is to analyze the empirical Rademacher complexity directly using convex duality as we have done for the linear and quadratic cases, and as we will do for the conic case next.

4.3.7 Complexity results with multiple conic constraints

Consider the function class

$$\mathcal{F} = \{f : f = \beta^T x, \beta^T \beta \leq B_b^2, \|A_k \beta\|_2 \leq a_k^T \beta + d_k \;\; \forall k = 1, ..., K\},$$

where we have one convex quadratic constraint and K conic constraints. We can find an upper bound on the Rademacher complexity as shown below.

Theorem 4.3.10. (Rademacher complexity of bounded linear function class with conic constraints) Let

$$\mathcal{F} = \{f : f = \beta^T x, \beta^T \beta \leq B_b^2, \|A_k \beta\|_2 \leq a_k^T \beta + d_k \;\; \forall k = 1, ..., K\},$$

where $B_b^2, \{A_k, a_k, d_k\}_{k=1}^K$ are the parameters. Assume $A_k \succ 0$ and let $\lambda_{\min}(A_k)$ denote

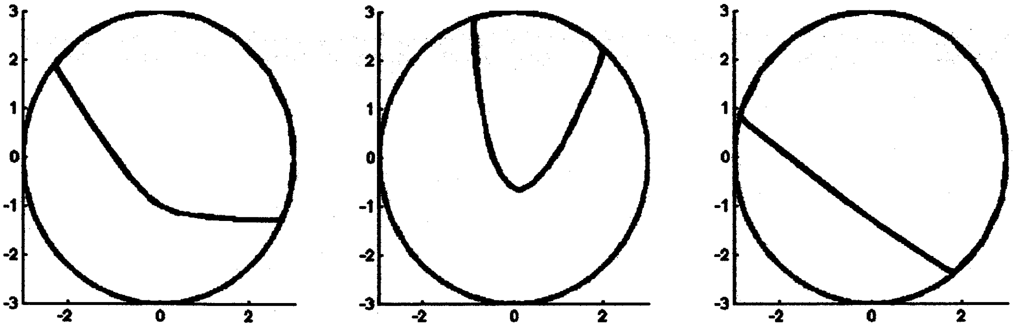


Figure 4-2: Here we illustrate the effect of a single conic constraint $\{\beta : \sqrt{4\mu\beta_1^2 + \mu\beta_2^2} \leq \delta(2\beta_1 + 3\beta_2 + 4)\}$ on our hypothesis space $\{\beta \in \mathbb{R}^2 : \beta^T\beta \leq 9\}$ for different scaling values of parameters μ and δ. In our notation, matrix $A = [2\sqrt{\mu}\ \ 0; 0\ \ \sqrt{\mu}]$, vector $a = \delta[2\ \ 3]^T$ and scalar $d = 4\delta$. *Left:* Parameter set (μ, δ) is equal to $(1, 1)$. The region covered by the conic constraint is the convex set in the upper part of the circle. *Center:* Changing the parameters (μ, δ) to $(10, 1)$ makes the eigenvalue $\lambda_{\min}(A)$ larger thus reducing the intersection region further. *Right:* Changing the parameters (μ, δ) to $(1, 10)$ increases the magnitude of $\|a\|_2$ and d relative to the value of $\lambda_{\min}(A)$ increases the intersection region between the conic constraint and the ball. This leads to a larger empirical Rademacher complexity bound value.

its minimum eigenvalue for $k = 1, ..., K$. Also let $\sup_{x \in \mathcal{X}} \|x\|_2 \leq X_b$. Then,

$$\bar{\mathcal{R}}(\mathcal{F}_{|S}) \leq \frac{X_b}{\sqrt{n}} \cdot \min\left\{B_b, \sum_{k=1}^{K} \frac{B_b\|a_k\|_2 + d_k}{K \cdot \lambda_{\min}(A_k)}\right\}.$$

Intuition: When $\|a_k\|_2$ and d_k are $o(\lambda_{\min}(A_k))$, the effect of conic constraints can influence the upper bound on the empirical Rademacher complexity and make the corresponding generalization bounds tighter. From a geometric point of view, we can infer the following: if the cones are sharp, then $\lambda_{\min}(A_k)$ are high, implying a smaller empirical Rademacher complexity. Figure 4-2 illustrates this in two dimensions.

Relation to standard results: The looser unconstrained version of the upper bound $\frac{X_b B_b}{\sqrt{n}}$ is recovered when there are no conic constraints or when the conic constraints are ineffective (for instance, when $\|a_k\|_2$ is high, d_k is a large offset or $\lambda_{\min}(A_k)$ is small).

Remark 4.3.11. There have been some recent attempts to obtain bounds on a related measure, similar to the empirical Gaussian complexity defined here, in the compressed sensing literature that also involves conic constraints Stojnic [2009]. Their objective (minimum number of measurements for signal recovery assuming sparsity) is very different from our objective (function class complexity and generalization). In the former context, there are a few results Chandrasekaran et al. [2012] dealing with the intersection of a single generic cone with a sphere ($\mathbb{S}^{p-1}$) whereas in this context, we look at the intersection of multiple second order cones (explicitly parameterized by $\{A_k, a_k, d_k\}_{k=1}^K$) with balls ($\{\beta^T\beta \leq B_b^2\}$).

4.4 Related Work

It is well-known that having additional unlabeled examples can aid in learning Shental et al. [2004], Nguyen and Caruana [2008b], Gómez-Chova et al. [2008], and this has been the subject of research in semi-supervised learning Zhu [2005]. The present work is fundamentally different than semi-supervised learning, because semi-supervised learning exploits the distributional properties of the set of unlabeled examples. In this work, we do not necessarily have enough unlabeled examples to study these distributional properties, but these unlabeled examples do provide us information about the hypothesis space. Distributional properties used in semi-supervised learning include cluster assumptions Singh et al. [2008], Rigollet [2007] and manifold assumptions Belkin and Niyogi [2004], Belkin et al. [2004]. In our work, the information we get from the unlabeled examples allows us to restrict the hypothesis space, which lets us be in the framework of empirical risk minimization and give theoretical generalization bounds via complexity measures of the restricted hypothesis spaces Bartlett and Mendelson [2002], Vapnik [1998]. While the focus of many works [e.g., Zhang, 2002, Maurer, 2006] is on complexity measures for ball-like function classes, our hypothesis spaces are more complicated, and arise here from constraints on the data.

Researchers have also attempted to incorporate domain knowledge directly into learning algorithms, where this domain knowledge does not necessarily arise from unlabeled examples. For instance, the framework of knowledge based SVMs Fung et al. [2002], Le et al. [2006] motivates the use of various constraints or modifications in the learning procedure to incorporate specific kinds of knowledge (without using unlabeled examples). The focus of Fung et al. [2002] is algorithmic and they consider linear constraints. Le et al. [2006] incorporate knowledge by modifying the function class itself, for instance, from linear function to non-linear functions.

In a different framework, that of Valiant's PAC learning, there are concentration statements about the risks in the presence of unlabeled examples Balcan and Blum [2005], Kääriäinen [2005], though in these results, the unlabeled points are used in a very different way than in our work. Specifically, in the work of Balcan and Blum [2005], the authors introduce the notion of incompatibility $\mathbb{E}_{x \sim D}[1 - \chi(h, x)]$ between a function h and the input distribution D. The unlabeled examples are used to estimate the distribution dependent quantity $\mathbb{E}_{x \sim D}[1 - \chi(h, x)]$. By imposing the constraint that models have their incompatibility with the distribution of the data source D below a desired level, we restrict the hypothesis space. Their result for a finite hypothesis space is as follows:

Theorem 4.4.1. [Theorem 1 of Balcan and Blum, 2005] If we see m unlabeled examples and n labeled examples, where

$$m \geq \frac{1}{\epsilon}\left[\ln |C| + \ln \frac{2}{\delta}\right] \text{ and } n \geq \frac{1}{\epsilon}\left[\ln |C_{D,\chi}(\epsilon)| + \ln \frac{2}{\delta}\right],$$

then with probability $1-\delta$, all $h \in C$ with zero training error and zero incompatibility $\frac{1}{m}\sum_{i=1}^{m}(1 - \chi(h, \tilde{x}_i)) = 0$, we have $\mathbb{E}[l(h(x), y)] \leq \epsilon$.

Here C is the finite hypothesis space of which h is an element and $C_{D,\chi}(\epsilon) = \{h \in C : \mathbb{E}_{x \sim D}[1 - \chi(h, x)] \leq \epsilon\}$. In the work of Kääriäinen [2005], the author obtains a generalization bound by approximating the disagreement probability of pairs of classifiers using unlabeled data. Again, here the unlabeled data is used to estimate a distribution dependent quantity, namely, the true disagreement probability between consistent models. In particular, the disagreement between two models h and g is defined to be $d(h, g) = \frac{1}{m}\sum_{i=1}^{m} 1_{[h(\tilde{x}_i) \neq g(\tilde{x}_i)]}$. The following theorem about generalization is proposed.

Theorem 4.4.2. Let $\mathcal{F}$ be the class of consistent models, that is, the set of models with zero training error. Assume the true model belongs to this class. Let $\hat{f} \in \mathcal{F}$ be the function whose distance to the farthest function in $\mathcal{F}$ is minimal (via metric d). Then, for all S, with probability $1-\delta$ over the choice of unlabeled sample $S^{\text{unlabeled}}$,

$$\begin{aligned}\mathbb{E}_{S^{\text{unlabeled}}}[l(\hat{f}(x), y)] \leq & \inf_{f \in \mathcal{F}} \sup_{g \in \mathcal{F}} d(f,g) \\ & + \bar{\mathcal{R}}(\{1_{[g \neq g']} | g, g' \in F\}_{|S^{\text{unlabeled}}}) + O\left(\sqrt{\frac{\ln(2/\delta)}{m}}\right).\end{aligned}$$

Note that the randomization in both Theorems 4.4.1 and 4.4.2 is also over unlabeled data. In our theorems, we do not randomize with respect to the unlabeled data. For us, they serve a different purpose and do not need to be chosen randomly. While their results focus on exploiting unlabeled data to estimate distribution dependent quantities, our technology focuses on exploiting unlabeled data to restrict the hypothesis space directly.

4.5 Proofs

4.5.1 Proof of Proposition 4.3.2

Proof. Instead of working with the maximization problem in the definition of empirical Rademacher complexity, we will work with a couple of related maximization problems, due to the following lemma.

Lemma 4.5.1.

$$\bar{\mathcal{R}}(\mathcal{F}_{|S}) \leq \mathbb{E}\left[\max\left(\sup_{f \in \mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i f(x_i), \sup_{f \in \mathcal{F}} -\frac{1}{n}\sum_{i=1}^{n} \sigma_i f(x_i)\right)\right]. \tag{4.4}$$

Proof. Since the empirical Rademacher complexity is defined as $\mathbb{E}_\sigma[\sup_{f\in\mathcal{F}} \frac{1}{n}|\sum_{i=1}^n \sigma_i f(x_i)|]$, we will show that for any fixed σ vector,

$$\sup_{f\in\mathcal{F}} \frac{1}{n}\left|\sum_{i=1}^n \sigma_i f(x_i)\right| \leq \max\left(\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \sigma_i f(x_i), \sup_{f\in\mathcal{F}} -\frac{1}{n}\sum_{i=1}^n \sigma_i f(x_i)\right). \tag{4.5}$$

The inequality above is straightforward to prove. Let f^* be the optimal solution to the maximization problem on the left. Then, f^* is a feasible point for each of the maximization problems on the right. We will look at two cases: In the first case, let $\frac{1}{n}\sum_{i=1}^n \sigma_i f^*(x_i) \geq 0$. Then, clearly the first maximization problem on the right, namely, $\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \sigma_i f(x_i)$ will have an optimal value greater than or equal to the left side of Equation (4.5). In the second case, let $\frac{1}{n}\sum_{i=1}^n \sigma_i f^*(x_i) < 0$. Then, the second maximization problem on the right, namely, $\sup_{f\in\mathcal{F}} -\frac{1}{n}\sum_{i=1}^n \sigma_i f(x_i)$ will have an optimal value greater than or equal to the left side of Equation (4.5). That is, in this case:

$$0 \leq \left|\frac{1}{n}\sum_{i=1}^n \sigma_i f^*(x_i)\right| = -\frac{1}{n}\sum_{i=1}^n \sigma_i f^*(x_i) \leq \sup_{f\in\mathcal{F}} -\frac{1}{n}\sum_{i=1}^n \sigma_i f(x_i).$$

Combining the two cases, we get the Equation (4.5). Taking expectations over σ gives us the desired inequality. □

Continuing with the proof of Proposition 4.3.2: Let $g = \sum_{i=1}^n \sigma_i x_i = X_L\sigma$ so that $\bar{\mathcal{R}}(\mathcal{F}_{|S}) = \frac{1}{n}\mathbb{E}[\sup_{\beta\in\mathcal{F}} |g^T\beta|]$. We will attempt to dualize the two maximization problems in the upper bound provided by Lemma 4.5.1 to get a bound on the empirical Rademacher complexity. Both maximization problems are very similar except for the objective. Let $\omega(g,\mathcal{F})$ be the optimal value of the following optimization problem:

$$\begin{aligned} &\max_\beta g^T\beta \;\; \text{s.t.} \\ &\beta^T\beta \leq B_b^2 \\ &a^T\beta \leq 1. \end{aligned}$$

Thus $\omega(g,\mathcal{F})$ represents the optimal value of the maximization problem inside the expectation operation in the first term of Equation (4.4). We will now write a dual program to the above and use weak duality to upper bound $\omega(g, \mathcal{F})$. The Lagrangian is:

$$\mathcal{L}(\beta, \gamma, \eta) = g^T\beta + \gamma(B_b^2 - \beta^T\beta) + \eta(1 - a^T\beta),$$

where $\beta \in \mathbb{R}^p, \gamma \in \mathbb{R}_+, \eta \in \mathbb{R}_+$. Maximizing the Lagrangian with respect to β gives us:

$$\begin{aligned}
&\max_{\beta} \mathcal{L}(\beta, \gamma, \eta) = \\
&\quad = \max_{\beta} \left[(g - \eta a)^T\beta - \gamma\beta^T\beta + \gamma B_b^2 + \eta\right] \\
&\quad = \max_{\beta} \left[-\gamma \left[\beta^T\beta - \frac{2(g - \eta a)^T\beta}{2\gamma} + \frac{\|g - \eta a\|_2^2}{4\gamma^2}\right] + \frac{\|g - \eta a\|_2^2}{4\gamma} + \gamma B_b^2 + \eta\right] \\
&\quad = \max_{\beta} \left[-\gamma \left\|\beta - \frac{g - \eta a}{2\gamma}\right\|_2^2 + \frac{\|g - \eta a\|_2^2}{4\gamma} + \gamma B_b^2 + \eta\right] \\
&\quad = \frac{\|g - \eta a\|_2^2}{4\gamma} + \gamma B_b^2 + \eta.
\end{aligned}$$

The dual problem is thus

$$\min_{\gamma \geq 0, \eta \geq 0} \frac{\|g - \eta a\|_2^2}{4\gamma} + \gamma B_b^2 + \eta.$$

Minimizing with respect to one of the decision variables, γ, gives the following dual problem

$$\min_{\eta \geq 0} B_b\|g - \eta a\|_2 + \eta.$$

Thus, $\omega(g, \mathcal{F}) \leq \min_{\eta \geq 0}(B_b\|g - \eta a\|_2 + \eta)$. Similarly we can prove an upper bound on the maximization problem appearing in the second term in the max operation in Equation (4.4), which will be $\min_{\eta \geq 0}(B_b\|g + \eta a\|_2 + \eta)$. Thus, the empirical Rademacher complexity is upper bounded as:

$$
\begin{aligned}
&\bar{\mathcal{R}}(\mathcal{F}_{|S})\\
&\le \frac{1}{n}\max\left(\mathbb{E}\left[\min_{\eta\ge 0}(B_b\|g-\eta a\|_2+\eta)\right],\mathbb{E}\left[\min_{\eta\ge 0}(B_b\|g+\eta a\|_2+\eta)\right]\right)\\
&=\frac{1}{n}\max\left(\mathbb{E}_\sigma\left[\min_{\eta\ge 0}(B_b\|X_L\sigma-\eta a\|_2+\eta)\right],\mathbb{E}_\sigma\left[\min_{\eta\ge 0}(B_b\|X_L\sigma+\eta a\|_2+\eta)\right]\right).
\end{aligned}
$$

□ □

4.5.2 Proof of Theorem 4.3.5

Proof. Consider the set $\mathcal{F}_{|S} = \{(\beta^T x_1, ..., \beta^T x_n) \in \mathbb{R}^n : \beta^T \mathbb{I}\beta \le B_b^2, \beta^T A_2 \beta \le 1\} \subset \mathbb{R}^n$. Let $\sigma = [\sigma_1, ..., \sigma_n]^T$. Also, let $\alpha = A_{\text{int}\gamma}^{1/2}\beta$.

$$
\begin{aligned}
\bar{\mathcal{R}}(\mathcal{F}_{|S}) &\overset{(a)}{\le}\frac{1}{n}\mathbb{E}_\sigma\left[\sup_{\{\beta:\beta^T A_{\text{int}\gamma}\beta\le 1\}}\left|\sum_{i=1}^{n}\sigma_i\beta^T x_i\right|\right]\\
&\overset{(b)}{=}\frac{1}{n}\mathbb{E}_\sigma\left[\sup_{\{\alpha:\alpha^T\alpha\le 1\}}\left|\sum_{i=1}^{n}\sigma_i(A_{\text{int}\gamma}^{-1/2}\alpha)^T x_i\right|\right]\\
&=\frac{1}{n}\mathbb{E}_\sigma\left[\sup_{\{\alpha:\|\alpha\|_2\le 1\}}\left|\alpha^T(A_{\text{int}\gamma}^{-1/2})^T X_L\sigma\right|\right]\\
&\overset{(c)}{=}\frac{1}{n}\mathbb{E}_\sigma\left[\|(A_{\text{int}\gamma}^{-1/2})^T X_L\sigma\|_2\right]\\
&\overset{(d)}{\le}\frac{1}{n}\sqrt{\mathbb{E}_\sigma\left[\|(A_{\text{int}\gamma}^{-1/2})^T X_L\sigma\|_2^2\right]}\\
&=\frac{1}{n}\sqrt{\mathbb{E}_\sigma\left[\text{trace}(X_L^T A_{\text{int}\gamma}^{-1} X_L\sigma\sigma^T)\right]}\\
&\overset{(e)}{=}\frac{1}{n}\sqrt{\text{trace}(X_L^T A_{\text{int}\gamma}^{-1} X_L)}
\end{aligned}
$$

where (a) follows because we are taking the supremum over the circumscribing ellipsoid; (b) follows because $A_{\text{int}\gamma}$ is positive definite, hence invertible; (c) is by Cauchy-Schwarz (equality case); (d) uses Jensen's inequality and (e) uses the linearity of trace and expectation to commute them along with the fact that $\mathbb{E}[\sigma\sigma^T] = I$. □ □

4.5.3 Proof of Theorem 4.3.7

Proof. Recall that we can decompose $A_{\text{int}\gamma}$ into a product $P^T DP$ where D is a diagonal matrix with the eigenvalues of $A_{\text{int}\gamma}$ as its entries and P is an orthogonal matrix (i.e., $P^T P = I$). Let us define a new variable: $\alpha := P\beta$, which is a linear transformation of linear model parameter β. Then, the empirical Gaussian complexity of our function class can be written as:

$$\bar{\mathcal{G}}(\mathcal{F}_{|S}) = \mathbf{E}_\sigma \left[\sup_{\alpha^T D\alpha \leq 1} \frac{1}{n} \sum_{i=1}^{n} \left| \sigma_i \alpha^T P x_i \right| \right],$$

where $\{\sigma_i\}_{i=1}^n$ are i.i.d. standard normal random variables. We now define a new vector ω to be a transformed version of the random vector $\sum_{i=1}^n \sigma_i x_i$. That is, let $\omega(\sigma) := P \sum_{i=1}^n \sigma_i x_i$. We will drop the dependence of ω on σ from the notation when it is clear from the context. The expression now becomes

$$n \cdot \bar{\mathcal{G}}(\mathcal{F}_{|S}) \geq \mathbf{E}_\sigma \left[\sup_{\alpha^T D\alpha \leq 1} \alpha^T \omega \right], \tag{4.6}$$

where the inequality is because we removed the absolute sign in the right hand side expression before substituting for ω.

The following are the major steps in our proof:

- We will analyze the Gaussian function $F(\omega(\sigma)) := \sup_{\alpha^T D\alpha \leq 1} \alpha^T \omega(\sigma)$ and show it is Lipschitz in σ. This is proved in Lemma 4.5.2.

- Then we apply Lemma 4.5.3, which is about Gaussian function concentration, to the above function. In particular, we will upper bound the variance of the Gaussian function $F(\omega(\sigma))$ in terms of its parameters (Lipschitz constant, matrix D, etc).

- We then generate a candidate lower bound for the empirical Gaussian complexity.

- The upper bound on the variance of $F(\omega(\sigma))$ we found earlier is used to make this bound proportional to $\sqrt{\text{trace}(X_L A_{\text{int}\gamma}^{-1} X_L)}$.

- Finally, we use a relation between empirical Rademacher complexity and empirical Gaussian complexity to obtain the desired result.

Computing a Lipschitz constant for $F(\omega(\sigma))$: The following lemma gives an upper bound on the Lipschitz constant of $F(\omega(\sigma))$.

Lemma 4.5.2. The function $F(\omega(\sigma)) := \sup_{\alpha^T D\alpha \leq 1} \alpha^T \omega(\sigma)$ is Lipschitz in σ with a Lipschitz constant $\mathcal{L}$ bounded by $X_b \sqrt{\frac{p \cdot n}{\lambda_{min}(D)}}$.

Proof. We have

$$F(\omega) = \sup_{\alpha^T D\alpha \leq 1} \alpha^T \omega = \sup_{(D^{1/2}\alpha)^T (D^{1/2}\alpha) \leq 1} \alpha^T \omega.$$

Using a new dummy variable $\rho = D^{1/2}\alpha$ we have:

$$F(\omega) = \sup_{\rho^T \rho \leq 1} (D^{-1/2}\rho)^T \omega = \sup_{\rho^T \rho \leq 1} \rho^T (D^{-1/2})^T \omega = \|D^{-1/2}\omega\|_2.$$

Thus,

$$\begin{aligned} |F(\omega_1) - F(\omega_2)| &= \left| \|D^{-1/2}\omega_1\|_2 - \|D^{-1/2}\omega_2\|_2 \right| \leq \|D^{-1/2}(\omega_1 - \omega_2)\|_2 \\ &\overset{(a)}{\leq} \left\| \frac{1}{\sqrt{\lambda_{min}(D)}} I(\omega_1 - \omega_2) \right\|_2 = \frac{1}{\sqrt{\lambda_{min}(D)}} \|\omega_1 - \omega_2\|_2. \end{aligned}$$

At (a), we used the fact that $D^{-1} \preceq \frac{1}{\lambda_{min}(D)} I$.

Now, we will upper bound $\|\omega_1 - \omega_2\|_2$ using σ_1 and σ_2 as follows. Using the definition of $\omega = PX_L\sigma$ we get,

$$\begin{aligned} \|\omega_1 - \omega_2\|_2 &= \|PX_L\sigma_1 - PX_L\sigma_2\|_2 = \|PX_L(\sigma_1 - \sigma_2)\|_2 \\ &\overset{(b)}{\leq} \|X_L(\sigma_1 - \sigma_2)\|_2 \\ &= \sqrt{(\sigma_1 - \sigma_2)^T X_L^T X_L (\sigma_1 - \sigma_2)} \end{aligned}$$

$$\overset{(c)}{\leq} \sqrt{(\sigma_1 - \sigma_2)^T \lambda_{max}(X_L^T X_L) I (\sigma_1 - \sigma_2)}$$
$$= \sqrt{\lambda_{max}(X_L^T X_L)} \|\sigma_1 - \sigma_2\|_2$$
$$\overset{(d)}{\leq} X_b \sqrt{p \cdot n} \|(\sigma_1 - \sigma_2)\|_2.$$

Here, (b) follows because P is an orthonormal matrix, (c) because $X_L^T X_L \preceq \lambda_{max}(X_L^T X_L) I$ and (d) because $\lambda_{max}(X_L^T X_L) \leq \text{trace}(X_L^T X_L) = \sum_{i=1}^n (X_L^T X_L)_{ii}$. Since, each diagonal element of $X_L^T X_L$ is a sum of p terms each upper bounded by X_b^2, we have $\lambda_{max}(X_L^T X_L) \leq n \cdot p \cdot X_b^2$. □ □

Upper bounding the variance of $F(\omega(\sigma))$ using Gaussian concentration: The following lemma describes concentration for Lipschitz functions of gaussian random variables.

Lemma 4.5.3. [Concentration, Tsirelson et al., 1976] If σ is a vector with i.i.d. standard normal entries and G is any function with Lipschitz constant $\mathcal{L}$ (with respect to the Euclidean norm), then

$$\mathbb{P}[|(G(\sigma) - \mathbb{E}[G(\sigma)]| \geq t] \leq 2e^{-\frac{t^2}{2\mathcal{L}^2}}.$$

The proof of Lemma 4.5.3 is omitted here. Using Lemmas 4.5.2 and 4.5.3 with $G(\sigma) = F(\omega)$, we have

$$\mathbb{P}[|(F(\omega) - \mathbb{E}_\sigma[F(\omega)]| \geq t] \leq 2e^{-\frac{t^2}{2\mathcal{L}^2}}, \tag{4.7}$$

where $\mathcal{L} = X_b \sqrt{\frac{p \cdot n}{\lambda_{min}(D)}}$.

Let $Y = |(F(\omega) - \mathbb{E}_\sigma[F(\omega)]|$. Then from the above tail bound, $P(Y^2 \geq s) \leq 2e^{-\frac{s}{2\mathcal{L}^2}}$ is true. Now we can bound the variance of $F(\omega)$ using the above inequality and the following lemma.

Lemma 4.5.4. For a random variable Y^2, $\mathbb{E}[Y^2] = \int_0^{+\infty} P(Y^2 \geq s) ds$.

Proof. This is an alternate expression for the expectation of a non-negative univariate random variable in terms of its distribution function. To show this, let us assume that the density function of Y^2 is μ_{Y^2}. We then have $P(Y^2 \geq s) = 1 - P(Y^2 \leq s) = 1 - \int_0^s \mu_{Y^2}(s')ds'$ and thus: $\mu_{Y^2}(s) = -\frac{dP(Y^2 \geq s)}{ds}$. So,

$$\begin{aligned}\mathbb{E}[Y^2] &= \int_0^{+\infty} s\mu_{Y^2}(s)ds = -\int_0^{+\infty} s\frac{dP(Y^2 \geq s)}{ds}ds \\ &= -[sP(Y^2 \geq s)]_0^{+\infty} + \int_0^{+\infty} P(Y^2 \geq s)ds.\end{aligned}$$

The first term is zero and we obtain our expression. □ □

The variance of $F(\omega)$, which is the same as the expectation of Y^2, can thus be upper bounded as follows:

$$\begin{aligned}\mathrm{Var}(F(\omega)) = \mathbb{E}_\sigma(Y^2) &\overset{(a)}{=} \int_0^{+\infty} P(Y^2 \geq s)ds \\ &\overset{(b)}{\leq} 2\int_0^{+\infty} e^{-\frac{s}{2\mathcal{C}^2}}ds = 4X_b^2\frac{p \cdot n}{\lambda_{min}(D)},\end{aligned} \tag{4.8}$$

where we used Lemma 4.5.4 for step (a) and Equation (4.7) for step (b) and finally substituting $X_b\sqrt{\frac{p \cdot n}{\lambda_{min}(D)}}$ for $\mathcal{C}$.

Lower bounding the empirical Gaussian complexity: Now we will lower bound the empirical Gaussian complexity by constructing a feasible candidate α' to substitute for the sup operation in Equation (4.6). Later, we will use the variance upper bound on $F(\omega)$ we found in the earlier section to make the bound more specific.

Let $j^* \in \{1, ..., p\}$ be the index at which the diagonal element $D(j^*, j^*) = \lambda_{min}(D)$. For each realization of σ (or equivalently ω) let $\alpha' = \left[0 \ldots \frac{|\omega_{j^*}|}{\omega_{j^*}\sqrt{\lambda_{min}(D)}} \ldots 0\right]$ with the non-zero entry at coordinate j^*. Clearly α' is a feasible vector in the ellipsoidal constraint $\{\alpha : \alpha^T D\alpha \leq 1\}$ seen in the complexity expression, Equation (4.6). Substituting it and using the definition of $F(\omega)$, we get a lower bound on the empirical

Gaussian complexity:

$$
\begin{aligned}
n \cdot \bar{\mathcal{G}}(\mathcal{F}_{|S}) \geq & \mathbb{E}_\sigma[F(\omega)] = \mathbb{E}_\sigma\left[\sup_{\alpha^T D\alpha \leq 1} \alpha^T\omega\right] \\
\overset{(a)}{\geq} & \mathbb{E}_\sigma[(\alpha')^T\omega] \overset{(b)}{\geq} \frac{1}{\sqrt{\lambda_{min}(D)}}\mathbb{E}_\sigma[|\omega_{j^*}|].
\end{aligned}
$$

Step (a) comes from the fact that α' is feasible in $\{\alpha : \alpha^T D\alpha \leq 1\}$ but not necessarily the maximum, and step (b) comes from the definition of α'.

Making the lower bound more specific using variance of $F(\omega(\sigma))$: Note that compared to the upper bound on the related Rademacher complexity obtained in Theorem 4.3.5, the dependence of empirical Gaussian complexity on $A_{\text{int}\gamma}$ is weak (only via $\lambda_{min}(D)$). We will use the variance of $F(\omega)$ to obtain a lower bound very similar to the upper bound in Equation (4.1). Rearranging the terms in the previous inequality, we get:

$$
\frac{(\mathbb{E}_\sigma[F(\omega)])^2}{(\mathbb{E}_\sigma|\omega_{j^*}|)^2} \geq \frac{1}{\lambda_{min}(D)}. \tag{4.9}
$$

By rewriting the variance in terms of the second and first moments, using expression (4.8) and then using (4.9) we get

$$
\begin{aligned}
\operatorname{Var}(F(\omega)) = & \mathbb{E}_\sigma[F^2(\omega)] - (\mathbb{E}_\sigma[F(\omega)])^2 \\
\leq & 4X_b^2 \frac{p \cdot n}{\lambda_{min}(D)} \leq 4pnX_b^2 \frac{(\mathbb{E}_\sigma[F(\omega)])^2}{(\mathbb{E}_\sigma|\omega_{j^*}|)^2}.
\end{aligned}
$$

Using expression (4.6) again, and then rearranging the terms in the previous expression, we obtain another lower bound on the scaled Gaussian complexity, which is:

$$
\left(n \cdot \bar{\mathcal{G}}(\mathcal{F}_{|S})\right)^2 \geq (\mathbb{E}_\sigma[F(\omega)])^2 \geq \frac{\mathbb{E}_\sigma[(F(\omega))^2]}{1 + \frac{4pnX_b^2}{(\mathbb{E}_\sigma|\omega_{j^*}|)^2}}
$$

$$=\frac{\mathbb{E}_\sigma[(\sup_{\alpha^T D\alpha\leq 1}\omega^T\alpha)^2]}{1+\frac{4pnX_b^2}{(\mathbb{E}_\sigma|\omega_{j^*}|)^2}}. \tag{4.10}$$

We can now try to bound two easier quantities $\mathbb{E}_\sigma[(\sup_{\alpha^T D\alpha\leq 1}\omega^T\alpha)^2]$ and $\mathbb{E}_\sigma|\omega_{j^*}|$ to get an expression for scaled Gaussian complexity and consequently for the empirical Rademacher complexity.

Let us start first with $\mathbb{E}|\omega_{j^*}|$. By definition ω equals $PX_L\sigma$. Thus, the j^*th coordinate of ω will be $\sum_i \sigma_i(Px_i)_{j^*}$ where $(\cdot)_{j^*}$ represents the j^*th coordinate of the vector. Since the σ_i are independent standard normal, their weighted sum ω is also standard normal with variance $\sum_i(Px_i)_{j^*}^2$. Since for any normal random variable z with mean zero and variance d it is true that $\mathbb{E}[|z|]=\sqrt{\frac{2d}{\pi}}$, we have

$$\begin{aligned}\mathbb{E}_\sigma[|w_{j^*}|] =&\sqrt{\frac{2}{\pi}}\left(\sum_i (Px_i)_{j^*}^2\right)^{\frac{1}{2}}\\ \geq&\sqrt{\frac{2}{\pi}}\min_{j=1,\ldots,p}\|(PX_L)_j\|_2 \end{aligned}\tag{4.11}$$

where $(PX_L)_j$ represents the j^{th} row of the matrix PX_L. For the second moment term of (4.10) that we need to bound, $\mathbb{E}_\sigma[(\sup_{\alpha^T D\alpha\leq 1}\omega^T\alpha)^2]$, we can see that

$$\begin{aligned}\sup_{\alpha^T D\alpha\leq 1}\omega^T\alpha =& \sup_{\tilde{\alpha}^T\tilde{\alpha}\leq 1}(PX_L\sigma)^T D^{-1/2}\tilde{\alpha}\\ =&\|D^{-1/2}PX_L\sigma\|_2.\end{aligned}$$

Thus,

$$\begin{aligned}\mathbb{E}_\sigma\left[\left(\sup_{\alpha^T D\alpha\leq 1}\omega^T\alpha\right)^2\right] =&\mathbb{E}_\sigma[\|D^{-1/2}PX_L\sigma\|_2^2]\\ =&\mathbb{E}_\sigma[(D^{-1/2}PX_L\sigma)^T D^{-1/2}PX_L\sigma]\\ =&\mathbb{E}_\sigma[\sigma^T X_L^T A_{\text{int}\gamma}^{-1}X_L\sigma]\\ =&\mathbb{E}_\sigma[\text{trace}(\sigma^T X_L^T A_{\text{int}\gamma}^{-1}X_L\sigma)]\\ =&\mathbb{E}_\sigma[\text{trace}(X_L^T A_{\text{int}\gamma}^{-1}X_L\sigma\sigma^T)]\end{aligned}$$

$$= \mathrm{trace}(X_L^T A_{\mathrm{int}\gamma}^{-1} X_L). \tag{4.12}$$

Substituting the two bounds we just derived, (4.11) and (4.12), into (4.10) gives us a lower bound on the scaled Gaussian complexity:

$$\left(n \cdot \bar{\mathcal{G}}(\mathcal{F}_{|S})\right)^2 \geq \frac{\mathrm{trace}(X_L^T A_{\mathrm{int}\gamma}^{-1} X_L)}{1 + \frac{4pnX_b^2}{(\sqrt{\frac{2}{\pi}} \min_{j=1,\ldots,p} \|(PX_L)_j\|_2)^2}}$$

$$n \cdot \bar{\mathcal{G}}(\mathcal{F}_{|S}) \geq \sqrt{\frac{\mathrm{trace}(X_L^T A_{\mathrm{int}\gamma}^{-1} X_L)}{1 + \frac{4pnX_b^2}{(\sqrt{\frac{2}{\pi}} \min_{j=1,\ldots,p} \|(PX_L)_j\|_2)^2}}}.$$

Using the relation between Rademacher and Gaussian complexities: The empirical Gaussian complexity is related to the empirical Rademacher complexity as follows.

Lemma 4.5.5. [Lemma 4 of Bartlett and Mendelson, 2002] There are absolute constants C and C' such that for every $\mathcal{F}_{|S}$ with $|S| = n$,

$$C'\bar{\mathcal{R}}(\mathcal{F}_{|S}) \leq \bar{\mathcal{G}}(\mathcal{F}_{|S}) \leq C \log(n) \bar{\mathcal{R}}(\mathcal{F}_{|S}).$$

Using the above result gives:

$$nC \log(n) \bar{\mathcal{R}}(\mathcal{F}_{|S}) \geq \sqrt{\frac{\mathrm{trace}(X_L^T A_{\mathrm{int}\gamma}^{-1} X_L)}{1 + \frac{4pnX_b^2}{(\sqrt{\frac{2}{\pi}} \min_{j=1,\ldots,p} \|(PX_L)_j\|_2)^2}}}$$

Thus, we get our desired result:

$$\bar{\mathcal{R}}(\mathcal{F}_{|S}) \geq \frac{\kappa}{n \log n} \sqrt{\mathrm{trace}(X_L^T A_{\mathrm{int}\gamma}^{-1} X_L)},$$

where

$$\kappa = \frac{1}{C\sqrt{1 + \frac{2\pi pnX_b^2}{(\min_{j=1,\ldots,p} \|(PX_L)_j\|_2)^2}}}.$$

□ □

4.5.4 Proof of Corollary 4.3.9

Proof. Since the ellipsoid defined using $A_{\text{int}\gamma}$ circumscribes the region of intersection of ellipsoids determined by A_1 and A_2, we have

$$\begin{aligned}\mathcal{F} = \Big\{f|f(x) =&\beta^T x, \beta \in \mathbb{R}^p, \beta^T A_1 \beta \leq 1, \beta^T A_2 \beta \leq 1, \\ &\sum_{j=1}^{p} c_{j\nu}\beta_j + \delta_\nu \leq 1, \delta_\nu > 0, \nu = 1, ..., V\Big\} \\ &\subseteq \\ \Big\{f|f(x) =&\beta^T x, \beta \in \mathbb{R}^p, \beta^T A_{\text{int}\gamma} \beta \leq 1, \\ &\sum_{j=1}^{p} c_{j\nu}\beta_j + \delta_\nu \leq 1, \delta_\nu > 0, \nu = 1, ..., V\Big\} =: \mathcal{F}'.\end{aligned}$$

Further, $\beta^T \lambda_{\min}(A_{\text{int}\gamma}) I \beta \leq \beta^T A_{\text{int}\gamma} \beta \leq 1$ since $\lambda_{\min}(A_{\text{int}\gamma}) I \preceq A_{\text{int}\gamma}$. That is, the set $\beta^T \lambda_{\min}(A_{\text{int}\gamma}) I \beta \leq 1$ is bigger than the ellipsoid defined using $A_{\text{int}\gamma}$. Thus,

$$\begin{aligned}\mathcal{F}' = \Big\{f|f(x) =&\beta^T x, \beta \in \mathbb{R}^p, \beta^T A_{\text{int}\gamma} \beta \leq 1, \\ &\sum_{j=1}^{p} c_{j\nu}\beta_j + \delta_\nu \leq 1, \delta_\nu > 0, \nu = 1, ..., V\Big\} \\ &\subseteq \\ \Big\{f|f(x) =&\beta^T x, \beta \in \mathbb{R}^p, \beta^T \beta \leq \frac{1}{\lambda_{\min}(A_{\text{int}\gamma})}, \\ &\sum_{j=1}^{p} c_{j\nu}\beta_j + \delta_\nu \leq 1, \delta_\nu > 0, \nu = 1, ..., V\Big\} =: \mathcal{F}''.\end{aligned}$$

Noting that $\beta^T \beta \leq \frac{1}{\lambda_{\min}(A_{\text{int}\gamma})}$ is the same as $\|\beta\|_2 \leq \sqrt{\lambda_{\max}(A_{\text{int}\gamma}^{-1})}$, we can use Theorem 4.3.3 on $\mathcal{F}''$ with $r = 2, q = 2$ and $\mathcal{B}_b := \sqrt{\lambda_{\max}(A_{\text{int}\gamma}^{-1})}$ to get a bound on $N(\sqrt{n}\epsilon, \mathcal{F}''_{|S}, \|\cdot\|_2) \geq N(\sqrt{n}\epsilon, \mathcal{F}_{|S}, \|\cdot\|_2)$ giving us the stated result. □ □

4.5.5 Proof of Theorem 4.3.8

Proof. Let $g = \sum_{i=1}^{n} \sigma_i x_i = X_L \sigma$ so that $\bar{\mathcal{R}}(\mathcal{F}_{|S}) = \frac{1}{n}\mathbb{E}[\sup_{\beta \in \mathcal{F}} |g^T \beta|]$. Instead of directly working with the empirical Rademacher complexity, we will dualize the two maximization problems in the upper bound given by Equation (4.4) of Lemma 4.5.1. Both maximization problems are very similar except for the objective. Let $\omega(g, \mathcal{F})$ be the optimal value of the following optimization problem:

$$\begin{aligned} \max_{\beta} g^T \beta \ \text{ s.t.} \\ \beta^T \beta \leq B_b^2 \\ \beta^T A_2 \beta \leq 1. \end{aligned}$$

Thus $\omega(g, \mathcal{F})$ is proportional to the first term inside the max operation in Equation (4.4), which gives an upper bound in the empirical Rademacher complexity. We will now write a dual program to the above and use weak duality to upper bound $\omega(g, \mathcal{F})$. The Lagrangian is:

$$\mathcal{L}(\beta, \gamma, \eta) = g^T \beta + \gamma(B_b^2 - \beta^T \beta) + \eta(1 - \beta^T A_2 \beta),$$

where $\beta \in \mathbb{R}^p, \gamma \in \mathbb{R}_+, \eta \in \mathbb{R}_+$. Maximizing the Lagrangian with respect to β gives us:

$$\begin{aligned} &\max_{\beta} \mathcal{L}(\beta, \gamma, \eta) = \\ &\quad = \max_{\beta} \left[g^T \beta - \gamma \beta^T \beta - \eta \beta^T A_2 \beta + \gamma B_b^2 + \eta \right] \\ &\quad = \max_{\beta} \left[- \left(-g^T \beta + \beta^T (\gamma \mathbf{I} + \eta A_2) \beta \right) + \gamma B_b^2 + \eta \right] \\ &\quad = \max_{\beta} \left[- \left(-g^T (\gamma \mathbf{I} + \eta A_2)^{-1/2} (\gamma \mathbf{I} + \eta A_2)^{1/2} \beta \right. \right. \\ &\qquad\quad \left. \left. + \beta^T (\gamma \mathbf{I} + \eta A_2)^{1/2} (\gamma \mathbf{I} + \eta A_2)^{1/2} \beta \right) + \gamma B_b^2 + \eta \right] \\ &\quad = \max_{\beta} \left[- \left\| (\gamma \mathbf{I} + \eta A_2)^{1/2} \beta - \frac{(\gamma \mathbf{I} + \eta A_2)^{-1/2} g}{2} \right\|_2^2 \right. \end{aligned}$$

$$
\left. + \frac{\|(\gamma \mathbb{I} + \eta A_2)^{-1/2} g\|_2^2}{4} + \gamma B_b^2 + \eta \right]
$$

$$
= \frac{\|(\gamma \mathbb{I} + \eta A_2)^{-1/2} g\|_2^2}{4} + \gamma B_b^2 + \eta,
$$

where in the last step we set $\beta = \frac{(\gamma \mathbb{I} + \eta A_2)^{-1} g}{2}$. The dual problem is thus:

$$
\min_{\gamma \geq 0, \eta \geq 0} \frac{\|(\gamma \mathbb{I} + \eta A_2)^{-1/2} g\|_2^2}{4} + \gamma B_b^2 + \eta, \text{ or equivalently,}
$$

$$
\min_{\gamma \geq 0, \eta \geq 0} \frac{1}{4} g^T (\gamma \mathbb{I} + \eta A_2)^{-1} g + \gamma B_b^2 + \eta.
$$

If we let $\gamma = 1 - \eta$, we are further constraining the minimization problem, yielding another upper bound of the form:

$$
\omega(g, \mathcal{F}) \leq \min_{\eta \in [0,1]} \frac{1}{4} g^T (\mathbb{I} + \eta (A_2 - \mathbb{I}))^{-1} g + B_b^2 + \eta (1 - B_b^2).
$$

If we consider the second maximization problem $\sup_{\beta \in \mathcal{F}} -g^T \beta$ that appears in Equation (4.4), we can similarly upper bound its optimal value with the same minimization problem as $\omega(g, \mathcal{F})$. One intuitive reason why the same minimization problem serves as an upper bound is because the hypothesis class $\mathcal{F}$ is closed under negation. Thus, we get an upper bound on the empirical Rademacher complexity as:

$$
\begin{aligned}
\bar{\mathcal{R}}(\mathcal{F}_{|S}) &\leq \mathbb{E}\left[\frac{1}{n} \omega(g, \mathcal{F})\right] \\
&\leq \mathbb{E}\left[\frac{1}{n} \min_{\eta \in [0,1]} \frac{1}{4} g^T (\mathbb{I} + \eta (A_2 - \mathbb{I}))^{-1} g + B_b^2 + \eta (1 - B_b^2)\right],
\end{aligned}
$$

where recall that $g = \sum_{i=1}^n \sigma_i x_i$. Fix any feasible η. Let $A_{\text{int}\eta} := (\mathbb{I} + \eta (A_2 - \mathbb{I}))$ (it corresponds to an ellipsoid as well since $\eta \in [0, 1]$). Then,

$$
\begin{aligned}
\bar{\mathcal{R}}(\mathcal{F}_{|S}) &\leq \mathbb{E}\left[\frac{1}{4n} \sigma^T X_L^T A_{\text{int}\eta}^{-1} X_L \sigma + \frac{1}{n} (B_b^2 + \eta (1 - B_b^2))\right] \\
&= \frac{1}{4n} \text{trace}(X_L^T A_{\text{int}\eta}^{-1} X_L) + \frac{1}{n} (B_b^2 + \eta (1 - B_b^2)).
\end{aligned}
$$

We can minimize the right hand side over $\eta \in [0,1]$ to get the desired result. □ □

4.5.6 Proof of Theorem 4.3.10

Proof. The core idea of the proof is to come up with an intuitive upper bound on the empirical Rademacher complexity of $\mathcal{F}$ using convex duality. We have already seen the use of convex duality in Proposition 4.3.2 and Theorem 4.3.8. Recall the definition of the empirical Rademacher complexity of a function class $\mathcal{F}$:

$$\bar{\mathcal{R}}(\mathcal{F}_{|S}) = \frac{1}{n}\mathbf{E}_\sigma\left[\sup_{\beta\in\mathcal{F}}\left|\sum_{i=1}^{n}\sigma_i(\beta^T x_i)\right|\right],$$

where $\{\sigma_i\}_{i=1}^n$ are i.i.d. Bernoulli random variables taking values in $\{\pm 1\}$ with equal probability. Now define a new vector g to be the random vector $\sum_{i=1}^n \sigma_i x_i$. As in the previous proofs, instead of directly working with the empirical Rademacher complexity, we will dualize the two maximization problems in the upper bound given by Equation (4.4) of Lemma 4.5.1. Let $\omega(g,\mathcal{F}) = \sup_{\beta\in\mathcal{F}} g^T\beta$. That is, $\omega(g,\mathcal{F})$ is the optimal value of the first maximization problem (ignoring factor $1/n$) appearing on the right hand side of Equation (4.4):

$$\begin{aligned}\max_{\beta}\ & g^T\beta \quad \text{s.t.}\\ & \beta^T\beta \le B_b^2\\ & \|A_k\beta\|_2 \le a_k^T\beta + d_k \ \ \forall k = 1,...,K.\end{aligned} \tag{4.13}$$

The Lagrangian of the problem can be written as Boyd and Vandenberghe [2004]:

$$\mathcal{L}(\beta,\gamma,\{z_k,\theta_k\}_{k=1}^K) = g^T\beta + \gamma(B_b^2 - \beta^T\beta) + \sum_{k=1}^{K}\left[z_k^T A_k\beta + \theta_k\cdot(a_k^T\beta + d_k)\right],$$

where $\beta \in \mathbb{R}^p, \gamma \in \mathbb{R}_+$ and for $k = 1,...,K$ we have $\|z_k\|_2 \le \theta_k$. For any set of feasible values of $(\beta,\gamma,\{z_k,\theta_k\}_{k=1}^K)$, the objective of the SOCP in Equation (4.13) is upper bounded by $\mathcal{L}(\beta,\gamma,\{z_k,\theta_k\}_{k=1}^K)$. Thus, $\omega(g,\mathcal{F}) \le \sup_\beta \mathcal{L}(\beta,\gamma,\{z_k,\theta_k\}_{k=1}^K)$. We will analyze this maximization problem as the first step towards a tractable bound on $\omega(g, \mathcal{F})$.

In the second step, we will minimize $\sup_\beta \mathcal{L}(\beta, \gamma, \{z_k, \theta_k\}_{k=1}^K)$ over variable γ (one of the dual variables) to get an upper bound on $\omega(g, \mathcal{F})$ in terms of $\{z_k, \theta_k\}_{k=1}^K$. These two steps are shown below:

First step: After rearranging terms and completing squares, we get the following dual objective to be minimized over dual variables γ and $\{z_k, \theta_k\}_{k=1}^K$.

$$\begin{aligned}
&\sup_{\beta \in \mathbf{R}^p} \mathcal{L}(\beta, \gamma, \{z_k, \theta_k\}_{k=1}^K) \\
&\quad = \sup_{\beta \in \mathbf{R}^p} \left[\left(g + \sum_{k=1}^K (A_k^T z_k + \theta_k a_k) \right)^T \beta + \gamma B_b^2 + \sum_{k=1}^K \theta_k d_k - \gamma \beta^T \beta \right] \\
&\quad = \sup_{\beta \in \mathbf{R}^p} \left[-\gamma \left\| \beta - \frac{g + \sum_{k=1}^K (A_k^T z_k + \theta_k a_k)}{2\gamma} \right\|_2^2 \right. \\
&\qquad \left. + \frac{\|g + \sum_{k=1}^K (A_k^T z_k + \theta_k a_k)\|_2^2}{4\gamma} + \left(\gamma B_b^2 + \sum_{k=1}^K \theta_k d_k \right) \right] \\
&\quad = \frac{\|g + \sum_{k=1}^K (A_k^T z_k + \theta_k a_k)\|_2^2}{4\gamma} + \gamma B_b^2 + \sum_{k=1}^K \theta_k d_k.
\end{aligned}$$

The second to last equality above is obtained by completing the squares (in terms of β) and the last equality is due to the fact that the optimal value is obtained when $\beta = \frac{g + \sum_{k=1}^K (A_k^T z_k + \theta_k a_k)}{2\gamma}$. The resulting term is now a function of the remaining variables (γ and $\{z_k, \theta_k\}_{k=1}^K$) and serves as an upper bound to $\omega(g, \mathcal{F})$ for any feasible values of γ and $\{z_k, \theta_k\}_{k=1}^K$.

Second step: Since $\min_{x,y} f(x, y) = \min_x(\min_y f(x, y))$ when $f(x, y)$ is convex and the feasible set is convex, we now minimize with respect to γ to get the following upper bound:

$$\begin{aligned}
&\inf_{\gamma \in \mathbf{R}_+} \sup_{\beta \in \mathbf{R}^p} \mathcal{L}(\beta, \gamma, \{z_k, \theta_k\}_{k=1}^K) \\
&\qquad = B_b \left\| g + \sum_{k=1}^K (A_k^T z_k + \theta_k a_k) \right\|_2 + \sum_{k=1}^K \theta_k d_k,
\end{aligned}$$

where the above statement follows because for a problem of the form $\min_{\gamma \in \mathbf{R}_+} \frac{a}{\gamma} + b\gamma + c$ with $a > 0, b > 0$, the optimal solution is $\gamma^* = +\sqrt{\frac{a}{b}}$.

Continuing, we now optimize over the remaining variables $\{z_k, \theta_k\}_{k=1}^K$ as follows:

$$\begin{aligned}\omega(g, \mathcal{F}) &= \sup_{\beta \in \mathcal{F}} g^T \beta \\ &\leq \inf_{\{(z_k, \theta_k): \|z_k\|_2 \leq \theta_k, k=1,..,K\}} B_b \left\| g + \sum_{k=1}^K (A_k^T z_k + \theta_k a_k) \right\|_2 + \sum_{k=1}^K \theta_k d_k. \qquad (4.14)\end{aligned}$$

An upper bound on $\omega(g, \mathcal{F})$ can be obtained by finding a set of optimal or feasible values for $\{z_k, \theta_k\}_{k=1}^K$. Note that since $A_k \succ 0$, $A_k^T = A_k$ and A_k^{-1} exists. Obtaining the optimal value of the minimization in Equation (4.14) is difficult analytically. Instead, we will pick a suitable feasible value for $\{z_k, \theta_k\}_{k=1}^K$. Plugging this feasible value will give us an upper bound on $\omega(g, \mathcal{F})$. In particular, let $z_k = -\frac{1}{K} A_k^{-1} g$. Then, setting $\theta_k = \frac{1}{K} \|A_k^{-1} g\|_2$ gives us a feasible value for each $\{z_k, \theta_k\}$. Thus,

$$\begin{aligned}\omega(g, \mathcal{F}) &\leq B_b \left\| g + \sum_{k=1}^K A_k^T \left(-\frac{1}{K} A_k^{-1} g \right) + \sum_{k=1}^K \frac{1}{K} \|A_k^{-1} g\|_2 a_k \right\|_2 + \sum_{k=1}^K \frac{1}{K} \|A_k^{-1} g\|_2 d_k \\ &= B_b \left\| g - g + \sum_{k=1}^K \frac{\|A_k^{-1} g\|_2}{K} a_k \right\|_2 + \sum_{k=1}^K \frac{\|A_k^{-1} g\|_2}{K} d_k \\ &= B_b \left\| \sum_{k=1}^K \frac{\|A_k^{-1} g\|_2}{K} a_k \right\|_2 + \sum_{k=1}^K \frac{\|A_k^{-1} g\|_2}{K} d_k \\ &\leq \sum_{k=1}^K \frac{\|A_k^{-1} g\|_2}{K} (B_b \|a_k\|_2 + d_k) \\ &\leq \|g\|_2 \sum_{k=1}^K \frac{B_b \|a_k\|_2 + d_k}{K \cdot \lambda_{\min}(A_k)}.\end{aligned}$$

Dualizing the second maximization problem in Equation (4.4) also gives us the same upper bound as obtained above for $\omega(g, \mathcal{F})$. That is, if $\omega'(g, \mathcal{F}) := \sup_{\beta \in \mathcal{F}} -g^T \beta$, then the same analysis as above (replacing g with $-g$) gives:

$$\omega'(g,\mathcal{F}) \le \|g\|_2 \sum_{k=1}^{K} \frac{B_b\|a_k\|_2 + d_k}{K \cdot \lambda_{\min}(A_k)}.$$

We can now come up with the desired upper bound for the empirical Rademacher complexity using Equation (4.4):

$$\begin{aligned}
\bar{\mathcal{R}}(\mathcal{F}_{|S}) &\le \mathbb{E}\left[\max\left(\frac{1}{n}\omega(g,\mathcal{F}), \frac{1}{n}\omega'(g,\mathcal{F})\right)\right] \\
&\le \frac{1}{n}\mathbb{E}\left[\|g\|_2 \sum_{k=1}^{K} \frac{B_b\|a_k\|_2 + d_k}{K \cdot \lambda_{\min}(A_k)}\right] \quad \text{(since upper bounds are the same)} \\
&= \frac{1}{n}\mathbb{E}_\sigma\left[\left\|\sum_{i=1}^{n} \sigma_i x_i\right\|_2\right] \sum_{k=1}^{K} \frac{B_b\|a_k\|_2 + d_k}{K \cdot \lambda_{\min}(A_k)} \\
&\le \frac{1}{n}\sqrt{\mathbb{E}_\sigma\left[\left\|\sum_{i=1}^{n} \sigma_i x_i\right\|_2^2\right]} \sum_{k=1}^{K} \frac{B_b\|a_k\|_2 + d_k}{K \cdot \lambda_{\min}(A_k)} \quad \text{(by Jensen's inequality)} \\
&\le \frac{X_b}{\sqrt{n}} \sum_{k=1}^{K} \frac{B_b\|a_k\|_2 + d_k}{K \cdot \lambda_{\min}(A_k)}.
\end{aligned}$$

In the case when there are no active conic constraints, we cannot use this bound. Instead, we can recover the well known standard bound by removing the terms related to conic constraints in Equation (4.14) and obtain only $\frac{X_b B_b}{\sqrt{n}}$. Combining both bounds we get,

$$\bar{\mathcal{R}}(\mathcal{F}_{|S}) \le \frac{X_b}{\sqrt{n}} \cdot \min\left\{B_b, \sum_{k=1}^{K} \frac{B_b\|a_k\|_2 + d_k}{K \cdot \lambda_{\min}(A_k)}\right\}.$$

□ □

4.6 Conclusion

In this chapter, we have outlined how various side information about a learning problem can effectively help in generalization. We focused our attention on several types of side information, leading to linear, polygonal, quadratic and conic constraints, giving motivating examples and deriving complexity measure bounds. This work goes beyond the traditional paradigm of ball-like hypothesis spaces to study more exotic, yet realistic, hypothesis spaces, and is a starting point for more work on other interesting hypothesis spaces.

Chapter 5

Robust Optimization using Machine Learning for Uncertainty Sets

5.1 Introduction

In this work, we consider a situation often faced by decision makers: a policy needs to be created for the future that would be a best possible reaction to the worst possible uncertain situation; this is a question of *robust optimization*. In our case, the decision maker does not know what the worst situation might be, and uses complex data to estimate the *uncertainty set*, which is the set of uncertain future situations. Here we are interested in answering questions such as: How might we construct a principled uncertainty set from these complex data? Can we ensure that with high probability our policy will be robust to whatever the future brings? Can we construct uncertainty sets that are useful for the situation at hand and are not too conservative?

In this chapter we address the important setting where detailed data (features) are available to predict each possible future situation. We turn to predictive modeling techniques from machine learning to make predictions, and to define uncertainty sets. Models created from finite data are uncertain: given a collection of historical data, there many be many predictive models that appear to be equally good, according to any measure of predictive quality. This was called the *Rashomon effect* by statistician Breiman [Breiman, 2001b], and it is this source of uncertainty in learning that we capture while designing uncertainty sets.

Our concept is possibly best explained through an illustrative example. Consider the minimum variance portfolio allocation problem where our goal is to construct a portfolio of assets. Let us temporarily say that we know exactly what the return for each of the assets in the market will be, and denote $\mathbf{y} \in \mathcal{Y} \subseteq \mathbb{R}^m$ as the vector of these known returns. Let the covariance of the returns be $\mathbf{\Sigma}$, which is also known in advance. We denote $\boldsymbol{\pi}$ as our choice of portfolio weights. We thus solve the basic decision-making problem:

$$\min_{\boldsymbol{\pi}} \boldsymbol{\pi}^T \mathbf{\Sigma} \boldsymbol{\pi} \quad \text{s.t. } \boldsymbol{\pi}^T \mathbf{1} = 1, \;\; \mathbf{y}^T \boldsymbol{\pi} \geq c,$$

where $()^T$ is the transpose operator, c is a constant and $\mathbf{1}$ is the vector of all ones. The objective represents the 'risk' of the portfolio that we wish to minimize and the two constraints represent that: (a) the sum of portfolio weights should be equal to one, and (b) the return on the portfolio should be lower bounded by an acceptable baseline rate of return denoted by c. Now let us consider the more realistic case where the returns $\mathbf{y}$ are not known in advance, and we need to make a decision about portfolio weights $\boldsymbol{\pi}$ under uncertainty (for simplicity of exposition, let us assume that $\mathbf{\Sigma}$ is known even though in reality we may need to estimate it along with the returns $\mathbf{y}$). If we are able to encode our uncertainty about these forecasted returns using an uncertainty set $\mathcal{U}$, then we can take a robust optimization (RO) approach and solve the following:

$$\min_{\boldsymbol{\pi}} \boldsymbol{\pi}^T \mathbf{\Sigma} \boldsymbol{\pi} \quad \text{s.t. } \boldsymbol{\pi}^T \mathbf{1} = 1, \;\; \mathbf{y}^T \boldsymbol{\pi} \geq c \, \forall \mathbf{y} \in \mathcal{U},$$

which gives us a best response to the worst possible outcome $\mathbf{y}$ in uncertainty set $\mathcal{U}$. The uncertainty set $\mathcal{U}$ can be defined in many ways, and the central goal of this work is how to model $\mathcal{U}$ from complex data from the past. These data take the form of features and labels; for instance in the portfolio allocation problem, the data are $\{(\mathbf{x}^i, \mathbf{y}^i)\}_{i=1}^n$ where an observation $\mathbf{x}^i \in \mathcal{X} \subseteq \mathbb{R}^d$ represents information we could use to predict the returns $\mathbf{y}^i \in \mathcal{Y}$ on past day i. These data might include macroeconomic indicators such as interest rates, employment statistics, retail sales and so on, as well as features of the assets themselves. Having complex data like this is very common, but often is not considered carefully within the decision problem. Some of the different ways uncertainty sets can be constructed are:

- Using a priori assumptions: We may have *a priori* knowledge about the range of possible future situations. In the portfolio allocation problem, we can assume that we know all possible values of the returns. This knowledge can guide us in constructing the returns uncertainty set $\mathcal{U}$ using interval constraints. That is, $\mathcal{U} := \{\mathbf{y} : \forall j, y_j \in [\underline{y}_j, \overline{y}_j]\}$, where we manually select $\underline{y}_j$ and $\overline{y}_j$ for each j. Here we ignore the complex past data altogether.
- Using empirical statistics: We could create an uncertainty set using empirical statistics of the data. In the portfolio allocation problem, we might define $\mathcal{U}$ to be the set of all return vectors that are close to return vectors $\mathbf{y}^i$ that have been realized in the past. Or, $\mathcal{U}$ could be the convex hull of past returns vectors. Here we ignore the $\mathbf{x}^i$'s altogether.
- Using linear regression to model complex data: Here, we use the complex past data $\{(\mathbf{x}^i, \mathbf{y}^i)\}_{i=1}^n$, but we make strong (potentially incorrect) assumptions on the probability distribution these data are drawn from. We use these assumptions to define a class of "good" predictive models $\mathcal{B}$ from $\mathcal{X} \to \mathcal{Y}$. Then, given a new feature vector $\tilde{\mathbf{x}}$ (also in $\mathcal{X}$), we use $\mathcal{B}$ to define an "intermediate" uncertainty set $\mathcal{U}_{\mathcal{B}}$ of all possible outcomes for each situation $\tilde{\mathbf{x}}$, and another "intermediate" uncertainty set $\mathcal{U}_{-\mathcal{B}}$ to capture model residuals. Together, these two sets can be used to define $\mathcal{U}$. This is illustrated for the portfolio allocation problem as follows.

We define $\mathcal{B}$ as all linear models $\beta : \mathcal{X} \to \mathcal{Y}$ that fall in the confidence interval determined using a linear regression fit under the usual normality assumption. We then define $\mathcal{U}_{\mathcal{B}}$ as predicted returns from these "good" models given a new feature vector $\tilde{\mathbf{x}}$. Additionally, using past data and normality assumptions, we can define the set of model residuals $\mathcal{U}_{-\mathcal{B}}$. Finally, $\mathcal{U}_{\mathcal{B}}$ and $\mathcal{U}_{-\mathcal{B}}$ are used to define the set $\mathcal{U}$ in the robust portfolio allocation formulation above. One should think of $\mathcal{U}_{\mathcal{B}}$ as including all predictions from all models that fit the data reasonably well with respect to the squared loss. And think of $\mathcal{U}_{-\mathcal{B}}$ as the union of prediction intervals around these models. Then, $\mathcal{U}$ is the union of the predictions and prediction intervals from all of the good models. This allows our decision to be robust to future realizations within any prediction interval from any reasonably good model. This approach uses all of the data, but makes strong, possibly untrue assumptions of normality.

- Using machine learning to model complex data, which is the topic of this work: This setting is more general than linear regression and with much weaker assumptions. Methods that make strong assumptions have limited applicability for modern datasets with thousands of features, and such assumptions may hinder prediction performance. In this work, we provide two principled ways to construct set $\mathcal{U}$ using historical data. We will present two methods for each approach. Both of these approaches use tools from statistical learning theory and make minimal assumptions about the data source. In particular:

(a) In the first approach, we optimize prediction models over the data $\{(\mathbf{x}^i, \mathbf{y}^i)\}_{i=1}^n$, and use them to construct uncertainty set $\mathcal{U}$. $\mathcal{U}$ is used within the robust optimization problem to construct $\boldsymbol{\pi}^*$, and Theorem 5.4.1 provides a guarantee on its robustness; this guarantee is derived using statistical learning theory. Theorem 5.4.1 describes the guarantee for a generic class of prediction models and Theorem 5.5.1 specializes the guarantee for a specific set of prediction models, namely, the conditional quantile models. Note that in this approach, we do not explicitly construct a set of "good" prediction models $\mathcal{B}$ as in the regression approaches discussed in the bullet point above; here $\mathcal{U}$ is defined only from the optimized prediction models and the new feature vector $\tilde{\mathbf{x}}$. The only assumption made in this approach is that the data are drawn i.i.d from an unknown source distribution. In particular, there is no normality assumption. Let us give examples of how the two methods we propose for this approach would work when $\mathcal{U}$ is constructed from a regression problem (like the portfolio setting discussed earlier):

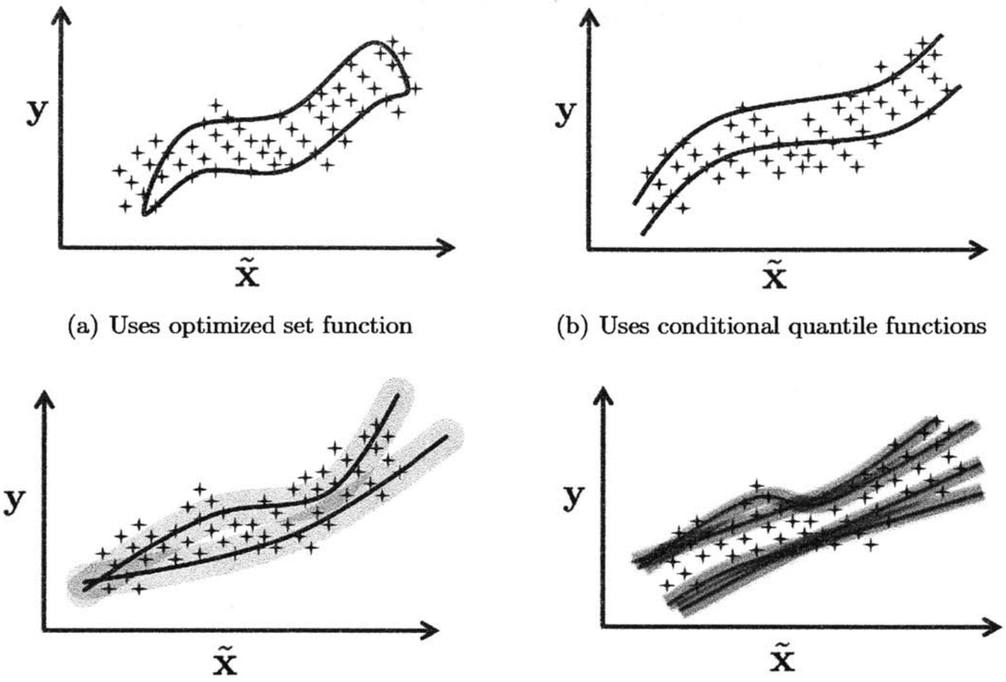
- For the first method, for every $\tilde{\mathbf{x}}$ the uncertainty set $\mathcal{U}$ corresponds to the domain of a indicator function on part of the set $\mathcal{Y}$. It is 1 on most of the training examples and is 0 farther away from them. Figure 5-1(a) shows an illustration of this.

- For the second method, we estimate the 95$^{\text{th}}$ and 5$^{\text{th}}$ percentiles of $\mathbf{y}$ given $\tilde{\mathbf{x}}$ and set $\mathcal{U}$ to be all values of $\mathbf{y} \in \mathcal{Y}$ between the two estimates. Figure 5-1(b) illustrates this.

(b) In the second approach, we consider the most extreme models within a class of "good" models $\mathcal{B}$. The set $\mathcal{B}$ contains all models within a parametric class that have low enough training error. We make only a single assumption: *with high probability, the error due to the 'best-in-class' model β^* is bounded with a known constant.* Our policies need to be robust to β^* that we would choose if we knew the distribution of data. Thus, we make efforts to ensure that the set of good models $\mathcal{B}$ that we will construct contains β^*. Here, $\mathcal{B}$ and $\mathcal{U}_{\mathcal{B}}$ are chosen in a distribution-independent manner, based on learning theory results. $\mathcal{U}_{-\mathcal{B}}$ is chosen based on our assumption on β^*. Theorems 5.6.1 and 5.7.1 give high probability guarantees on the robust optimal solution obtained using uncertainty set $\mathcal{U}$ constructed in this way. Theorem 5.6.1 corresponds to the case where a single prediction model is considered and Theorem 5.7.1 corresponds to the situation where two prediction models (for different quantiles) are considered. These guarantees are qualitatively different from the ones obtained in the first approach. To provide intuition for the two methods proposed for this approach in a regression setting (for instance, as in the portfolio problem):

- The third method would set $\mathcal{B}$ to be all elements of the hypothesis space (functions on $\mathcal{X} \mapsto \mathcal{Y}$) that have a low least squares loss on the dataset $\{(\mathbf{x}^i, \mathbf{y}^i)\}_{i=1}^n$. These functions estimate the mean of $\mathbf{y}$ given $\tilde{\mathbf{x}}$. Then we would take an interval above and below each element of $\mathcal{B}$. The union of those intervals would be the uncertainty set $\mathcal{U}$. Figure 5-1(c) illustrates this.

- The fourth method would set $\mathcal{B}^{0.95}$ to be all models of the 95th percentile of $\mathbf{y}$ given $\tilde{x}$ that have low loss. It would set $\mathcal{B}^{0.05}$ to be all models estimating the 5th percentile of $\mathbf{y}$ given $\tilde{x}$ that have low loss. We take an interval above and below



(a) Uses optimized set function

(b) Uses conditional quantile functions

(c) Uses an intermediate set of "good" models

(d) Uses two intermediate sets of "good" models

Figure 5-1: The empirical data $\{\mathbf{x}^i, \mathbf{y}^i\}_{i=1}^n$ is shown along with the boundaries created by the proposed methods in each of the above figures. Evaluation of these boundaries at a given $\tilde{\mathbf{x}}$ produces an uncertainty set. In (a), a set function is optimized over the sample and its evaluation at every $\tilde{\mathbf{x}}$ is plotted. In (b), we use optimized conditional quantile models to get the boundaries. In (c), we use an intermediate set of good prediction models and assumptions about model residuals to get the boundaries. In (d), we use two intermediate sets of good conditional quantile models. The lower and upper limits are used to define the boundaries.

each estimate provided by $\mathcal{B}^{0.95}$ and $\mathcal{B}^{0.05}$, and take the union of all of these intervals to form $\mathcal{U}$. Note that the fourth method is strictly more conservative than the second method the way we described it. Figure 5-1(d) illustrates this.

Being able to define uncertainty sets from predictive models is important: the uncertainty sets can now be specialized to a given new situation $\tilde{\mathbf{x}} \in \mathcal{X}$, and this is true even if we have never seen $\tilde{\mathbf{x}}$ before. For instance, when ordering daily supplies $\mathbf{y}^i$ for an ice cream parlor in Boston, an uncertainty set that depends on the weather might be much smaller than one that does not; planning for too much uncertainty in the weather can be too conservative and very costly: it would not be wise to budget for the largest possible summer sales in the middle of the winter. Though there have been attempts to define uncertainty sets in the linear regression setting [Goldfarb and Iyengar, 2003], ours is the first attempt to tackle the more general setting in a principled way.

Our goals are twofold: (i) We would like to create uncertainty sets for the more general machine learning setting using our proposed approaches (a) and (b) listed above. (ii) We would like to compute *sample complexity* values. That is, we want to determine how much data the practitioner needs for a guarantee that their chosen policy will be robust to future realizations. We provide finite sample guarantees on the quality of robustness using learning theory for both proposed approaches.

Our approaches for constructing uncertainty sets are flexible, intuitive, easy to understand from a practitioner's point of view, and at the same time can bring all the rich theoretical results of learning theory to justify the data-driven methodology. Our uncertainty set designs can handle prediction models for classification, regression, ranking and other supervised learning problems. A main theme of this work is that RO is a new context in which many learning theory results naturally apply and can be directly used.

In Section 5.3, we formulate our problem and discuss the two approaches (a) and (b) for making decisions under learning uncertainty. In Sections 5.4, 5.5, 5.6 and 5.7, we use learning theory techniques to justify the proposed uncertainty sets and state our probabilistic guarantees. Section 5.8 provides proofs for these guarantees. Finally, we conclude in Section 5.9.

5.2 Background Literature

There are many approaches to decision making under uncertainty when the uncertainty is due to finite data. Robustness is achieved either by taking into the uncertainty in the decision making formulation (as in RO discussed below), or by building robust statistical estimators [see Frost and Savarino, 1986, Jorion, 1986, for applications to portfolio problems].

In the optimization literature, there has been a continued interest in modeling uncertainty sets for robust optimization (RO) using empirical statistics of data [Delage and Ye, 2010], along with (strong) a priori assumptions about the probability distribution generating the parameters of a particular model for the data. Bertsimas et al. [2013] explore a way to specify data-driven uncertainty sets with probabilistic guarantees, where statistical hypothesis testing is used to construct sets. This approach is different from our approach in three important ways: (i) the method is designed for non-complex featureless data, (ii) the goal is totally different: For Bertsimas et al. [2013], the goal is to minimize the difference between the cost from a policy created using the true distribution and the cost from a policy from the estimated distribution, and (iii) our analysis based on learning theory [Vapnik, 1998] whereas their analysis is based on the theory of hypothesis testing. For us, the objective is to evaluate the feasibility of our policy with respect to a realization of the randomness in the future. The definition of "robustness" between our work and theirs is thus entirely different.

The closest work to ours is possibly that of Goldfarb and Iyengar [2003], who provide a linear-regression-based robust decision making paradigm for portfolio allocation problems, where they assume a multivariate linear regression model for the learning step. A big departure from this approach is that in our work, we are able to design uncertainty sets for a general class of decision making problems while making weak assumptions about the distributional aspects of the historical data. We base our uncertainty set design on regularized empirical risk minimization, which is quite a bit more general than regression. We contrast the sets constructed by Goldfarb and Iyengar [2003] with our proposed sets in Section 5.6.3.

Our work has the same flavor as *chance constrained programming* [Charnes and Cooper, 1959] and various other *stochastic programming* techniques. Both stochastic programming and robust optimization have extensions, for instance, for multi-stage decision making. We focus on single stage optimization. In our previous work [Tulabandhula and Rudin, 2013, 2014] we considered statistical learning theory bounds also for cases when unlabeled points were available. In that work, we considered prior knowledge about the outcome of an optimization problem that uses the $\tilde{y}^j$s. We showed that this kind of prior knowledge can create better generalization guarantees. Here, instead we study feasibility of the $\tilde{y}^j$s.

5.3 Formulation

In Sections 5.3.1 and 5.3.2 we will describe four ways to construct uncertainty set $\mathcal{U}$ using historical data and solve the corresponding robust optimization problems. The first two methods correspond to approach (a) in the introduction (Section 5.1), and the last two methods correspond to approach (b).

Let all the uncertain parameters of the decision problem be denoted by a vector $\mathbf{u} \in \mathbb{R}^m$. Given a realization of $\mathbf{u}$, let the (basic non-robust) decision making problem be written as:

$$\min_{\boldsymbol{\pi}} \rho(\boldsymbol{\pi}, \mathbf{u}) \quad \text{s.t.} \quad F(\boldsymbol{\pi}, \mathbf{u}) \in \mathcal{K}. \tag{5.1}$$

Here $\boldsymbol{\pi} \in \Pi \subseteq \mathbb{R}^{d_1}$ is the decision vector and $f : \Pi \times \mathbb{R}^m \to \mathbb{R}$ is the objective function. Function $F : \Pi \times \mathcal{U} \to \mathcal{K}$ and convex cone $\mathcal{K} \subseteq \mathbb{R}^{d_2}$ describe the constraints of the problem.

The robust version of the decision problem in Equation (5.1) is thus:

$$\min_{\boldsymbol{\pi}} \max_{\mathbf{u} \in \mathcal{U}} f(\boldsymbol{\pi}, \mathbf{u}) \quad \text{s.t.} \quad F(\boldsymbol{\pi}, \mathbf{u}) \in \mathcal{K} \text{ for all } \mathbf{u} \in \mathcal{U}, \tag{5.2}$$

where $\mathcal{U} \subset \mathbb{R}^m$ represents the uncertainty set. In Section 5.1, the minimum variance portfolio allocation problem is a specific instance of the decision problem in Equation (5.1). The robust portfolio allocation problem is an instantiation of the robust formulation in Equation (5.2).

To solve Equation (5.2), we prescribe the following steps:

Step 1: Construct $\mathcal{U}$ using any of the four methods listed in this section.

Step 2: Obtain a robust solution, using either of the two options below:

Option 1: If $\mathcal{U}$ is a "nice" set, then there are natural ways [Ben-Tal et al., 2009] to transform it into a relaxed set $\mathcal{U}'$ so that the robust optimization problem can be solved to obtain a robust solution π^*. For instance, if U can be bounded using a box or an ellipsoid, that box or ellipsoid can be $\mathcal{U}'$. If Equation 5.2 is a semi-infinite formulation that can be transformed into a finite formulation, then the finite formulation can be solved.

Option 2: If $\mathcal{U}$ is not a "nice" set, then do the following: sample L elements from $\mathcal{U}$ uniformly. For instance, this can be done using geometric random walks [e.g., Vempala, 2005] if $\mathcal{U}$ is convex. Then solve the sampled version of Equation (5.2) to obtain a robust solution π^* [see Calafiore and Campi, 2005] - this method assumes we have an efficient procedure to sample from $\mathcal{U}$.

We focus on **Step 1**. The goal is to ensure that the true realization of parameter $\mathbf{u} \in \mathbb{R}^m$ belongs to set $\mathcal{U}$ with a high likelihood. Let $\mathbf{u}$ be equal to an m-dimensional vector of unknown labels $[\tilde{y}^1 \ \ldots \ \tilde{y}^m]^T$, where each label $\tilde{y}^j \in \mathcal{Y}$ can be predicted given a corresponding feature vector $\tilde{\mathbf{x}}^j \in \mathcal{X}$. Thus m labels $\{\tilde{y}^j\}_{j=1}^m$, which can be forecasted from $\{\tilde{\mathbf{x}}^j\}_{j=1}^m$, feed into the decision problem of Equation (5.2).

In both approaches we propose, we will define $\mathcal{U}$ to be a product of m sets, each one constructed such that it contains the corresponding unknown true realization $\tilde{y}^j$ with high probability. Set $\mathcal{U}$ will be a function of training data sample $S = \{\mathbf{x}^i, y^i\}_{i=1}^n$ and the current feature vectors $\{\tilde{\mathbf{x}}^j\}_{j=1}^m$.

5.3.1 Direct use of empirically optimal prediction models

In this approach, we use empirically optimal prediction models directly. We start by discussing a very general form of prediction model, then discuss quantile regression.

General prediction models:

Let $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^d$ represent a feature vector and $y \in \mathcal{Y} \subseteq \mathbb{R}$ represent a label. Consider a class of set functions $I \in \mathcal{I}$, where $I : \mathcal{X} \to \mathfrak{M}_{\mathbb{R}}$, where $\mathfrak{M}_{\mathbb{R}}$ is the set of all measurable sets of $\mathbb{R}$. Let us say that we have a procedure that picks a function I^{Alg} so that most of the labels of the training examples obey $y^i \in I^{\mathrm{Alg}}(\mathbf{x}^i)$, $i = 1, ..., n$. As long as I^{Alg} belongs to a set of "simple" functions, we have a guarantee on how well I^{Alg} will generalize to new observations. Specifically, consider the following empirical risk minimization procedure:

$$\min_{I \in \mathcal{I}} \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}[y^i \notin I(\mathbf{x}^i)], \tag{5.3}$$

where $\mathbf{1}[\cdot]$ is the indicator function. Let an optimal solution to the above problem be I^{Alg}. Then, define the uncertainty set $\mathcal{U}$ as:

$$\mathcal{U} = \Pi_{j=1}^{m} I^{\text{Alg}}(\tilde{\mathbf{x}}^j), \tag{5.4}$$

where $\mathcal{U}$ is a product of m measurable sets. Figure 5-2(a) illustrates this construction in one dimension. Given this construction, **Step 1** of the workflow we described can be summarized as:

(a) Solve Equation (5.3) to obtain a set function I^{Alg} that depends on sample S.

(b) Define $\mathcal{U}$ according to Equation (5.4) using new observations $\{\tilde{\mathbf{x}}^j\}_{j=1}^{m}$.

The above setting is quite general. In particular, since the range of function I^{Alg} is $\mathfrak{M}_{\mathbf{R}}$, we can capture sets that are arbitrarily more complicated than simple intervals. For instance, if $\mathbb{P}_{y^j|\tilde{\mathbf{x}}^j}$ is bimodal, then for certain values of $\tilde{\mathbf{x}}^j$, $I^{\text{Alg}}(\tilde{\mathbf{x}}^j)$ can be the union of two disjoint intervals.

We remark that one can also approximate the source distribution $\mathbb{P}_{\mathbf{x},y}$ using an empirical distribution $\hat{\mathbb{P}}_{\mathbf{x},y}$ (there are many parametric and non-parametric ways to do this) and then construct set $\mathcal{U}$ using marginal distributions $\{\hat{P}_{y^j|\tilde{\mathbf{x}}^j}\}_{j=1}^{m}$. This would be slightly different than the approach described above in that it would require density estimation, which may itself be a hard problem. In the above method and the other methods below that we propose, we focus on estimating functionals of the conditional distributions $\{\hat{P}_{y^j|\tilde{\mathbf{x}}^j}\}$ directly.

Conditional quantile models:

In this method, we specialize the generic function class $\mathcal{I}$ to the class of set functions defined using conditional quantile models. We will estimate an upper quantile of $\tilde{y}$ for each $\tilde{\mathbf{x}}$, and a lower quantile of $\tilde{y}$ for each $\tilde{\mathbf{x}}$. The uncertainty set will be the interval between the two quantile estimates. This method is applicable when our prediction task is a regression problem.

When $y \sim \mathbb{P}_y$, the τ^{th} quantile of y, denoted by μ^τ, is defined as $\mu^\tau := \inf\{\mu : \mathbb{P}_y(y \le \mu) = \tau\}$. Here τ can vary between 0 and 1. In the special case when τ is set to 0.5, this defines the median. Similarly, when $(\mathbf{x}, y) \sim \mathbb{P}_{\mathbf{x},y}$, the conditional quantile μ^τ can be defined as a function from $\mathcal{X}$ to $\mathcal{Y}$, $\mu^\tau(\mathbf{x}) := \inf\{\mu : \mathbb{P}_{y|\mathbf{x}}(y \le \mu) = \tau\}$.

In our setting, $\tilde{y}^j$ conditioned on $\tilde{\mathbf{x}}^j$ is distributed according to $\mathbb{P}_{\tilde{y}^j|\tilde{\mathbf{x}}^j}$. Thus, given a value of $\tau \in [0,1]$, $\mathbb{P}_{\tilde{y}^j|\mathbf{x}=\tilde{\mathbf{x}}^j}(\tilde{y}^j \le \mu^\tau(\tilde{\mathbf{x}}^j)) = \tau$ where $\mu^\tau(\mathbf{x})$ is the conditional quantile defined earlier. Our method picks two values of τ, $\delta_p \le \delta_q$ such that:

$$\mathbb{P}_{\tilde{y}^j|\tilde{\mathbf{x}}^j}(\tilde{y}^j \le \mu^{\delta_p}(\tilde{\mathbf{x}}^j)) = \delta_p, \quad \text{and} \quad \mathbb{P}_{\tilde{y}^j|\tilde{\mathbf{x}}^j}(\tilde{y}^j \le \mu^{\delta_q}(\tilde{\mathbf{x}}^j)) = \delta_q.$$

For example, a typical value for the pair (δ_p, δ_q) can be $(0.05, 0.95)$ which makes $\mu^{\delta_p}(\tilde{\mathbf{x}}^j)$ correspond to the 5% conditional quantile and $\mu^{\delta_q}(\tilde{\mathbf{x}}^j)$ correspond to the 95% conditional quantile. Given these two conditional quantiles, we have:

$$\mathbb{P}_{\tilde{y}^j|\tilde{\mathbf{x}}^j}(\mu^{\delta_p}(\tilde{\mathbf{x}}^j) < \tilde{y}^j \le \mu^{\delta_q}(\tilde{\mathbf{x}}^j)) = \delta_q - \delta_p.$$

Thus, the unknown future realization of $\tilde{y}^j$ belongs to the interval $[\mu^{\delta_p}(\tilde{\mathbf{x}}^j), \mu^{\delta_q}(\tilde{\mathbf{x}}^j)]$ with high probability if δ_p and δ_q are chosen appropriately. If we knew the true conditional quantiles (which we do not), we could define the uncertainty set $\mathcal{U}$ as $\mathcal{U} = \Pi_{j=1}^m[\mu^{\delta_p}(\tilde{\mathbf{x}}^j), \mu^{\delta_q}(\tilde{\mathbf{x}}^j)]$. We will circumvent this issue by using sample $S = \{(\mathbf{x}^i, y^i)\}_{i=1}^n$ and quantile regression to obtain empirical quantile functions.

Quantile regression can be seen as an empirical risk minimization algorithm where the loss function is defined appropriately to obtain a conditional quantile function. That is, we aim to obtain an estimator function $\beta(\mathbf{x})$ of the true conditional quantile function $\mu^\tau(\mathbf{x})$ given a predefined quantile parameter τ. In particular, the *pinball* loss (or newsvendor loss) function defined below is used.

$$l^\tau(\beta(\mathbf{x}), y) = \begin{cases} \tau \cdot (y - \beta(\mathbf{x})) & \text{if } y - \beta(x) \geq 0, \\ (\tau - 1) \cdot (y - \beta(\mathbf{x})) & \text{otherwise.} \end{cases}$$

Let $l^\tau_{\mathbf{P}}(\beta) = \mathbb{E}_{\mathbf{x},y}[l^\tau(\beta(\mathbf{x}), y)]$. It can be shown [Koenker, 2005, Takeuchi et al., 2006] under some regularity conditions that the true conditional quantile function $\mu^\tau(\mathbf{x})$ is the minimizer of $l^\tau_{\mathbf{P}}(\beta)$ when minimized over all measurable functions. There are several works that consider linear and nonparametric quantile estimates using this loss function [Takeuchi et al., 2006, Rudin and Vahn, 2014]. In our setting, we will let $\mathcal{B}_0$ be our hypothesis class that we want to pick conditional quantile functions from.

Let the empirical risk minimization procedure using the pinball loss output a conditional quantile model $\beta^{Alg,\tau}$ when given the historical sample $S = \{(\mathbf{x}^i, y^i)\}_{i=1}^n$ of size n and a parameter τ. That is, let $l^\tau_S(\beta) = \frac{1}{n}\sum_{i=1}^n l^\tau(\beta(\mathbf{x}^i), y^i)$ and $\beta^{Alg,\tau} \in \arg\min_{\beta \in \mathcal{B}_0} l^\tau_S(\beta)$. The following definition of $\mathcal{U}$ uses two empirical conditional quantile functions with $\tau = \delta_p$ and $\tau = \delta_q$ respectively:

$$\mathcal{U} = \Pi_{j=1}^m \left[\min\left(\beta^{\text{Alg},\delta_p}(\tilde{\mathbf{x}}^j), \beta^{\text{Alg},\delta_q}(\tilde{\mathbf{x}}^j)\right), \max\left(\beta^{\text{Alg},\delta_p}(\tilde{\mathbf{x}}^j), \beta^{\text{Alg},\delta_q}(\tilde{\mathbf{x}}^j)\right)\right]. \tag{5.5}$$

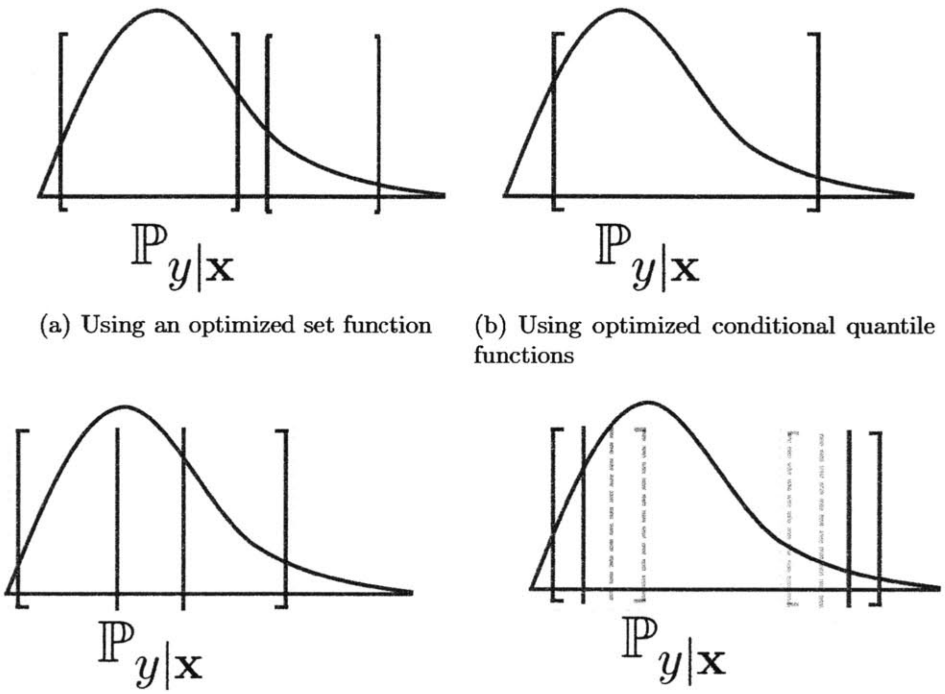
Here $\mathcal{U}$ is again a product of m intervals, each one constructed so that it contains the unknown $\tilde{y}^j$ with high probability (which we prove later). Figure 5-2(b) illustrates this construction in one dimension. Thus, for **Step 1**, we do the following:

1. Compute $\beta^{\text{Alg},\delta_p}$ and $\beta^{\text{Alg},\delta_q}$ using quantile regression.

2. Set $\mathcal{U}$ according to Equation (5.5).

5.3.2 Uncertainty set using an intermediate set of "good" prediction models

In this approach, we use optimized prediction models to define an intermediate set of "good" prediction models, which is then used to define $\mathcal{U}$. This approach aims


$\mathbb{P}_{y|\mathbf{x}}$
$\mathbb{P}_{y|\mathbf{x}}$
$\mathbb{P}_{y|\mathbf{x}}$
$\mathbb{P}_{y|\mathbf{x}}$

(a) Using an optimized set function

(b) Using optimized conditional quantile functions

(c) Using a single intermediate set of "good" models

(d) Using two intermediate sets of "good" models

Figure 5-2: The conditional distribution of y given $\mathbf{x}$ is shown along with the proposed uncertainty sets in each of the above figures. In (a), we use an optimized set function to directly define the subset of $\mathbb{R}$ that contains y with high probability. In (b), we use optimized conditional quantile models (the ones achieving the lowest training error) to directly define the set which contains the random variable y with high probability. In (c), we use an intermediate set of good prediction models to create $\mathcal{U}_{\mathcal{B}}$ and then enlarge the interval using set $\mathcal{U}_{-\mathcal{B}}$. In (d), we use two intermediate sets of good conditional quantile models and enlarge the corresponding intervals. The lower and upper limits of the two sets are then used to define $\mathcal{U}$.

to capture uncertainty in the modeling procedure explicitly: rather than using one predictive model, we use predictions from all models that we consider to be "good" with respect to our training data.

Using a single set of "good" prediction models:

Let $\beta : \mathcal{X} \mapsto \mathcal{Y}$ be a prediction model in the hypothesis class $\mathcal{B}_0$. For instance, $\mathcal{B}_0$ can be the set of linear predictors $\mathcal{B}_0 = \{x \mapsto \beta^T x : \|\beta\| \leq B_b\}$. Let $l(\beta(\mathbf{x}), y)$ denote the loss function. For example, $(\beta(\mathbf{x}) - y)^2$ is the least squares loss and $[1 - \beta(\mathbf{x})y]_+$ is the hinge loss used in Support Vector Machines. For any given model, let $l_{\mathbb{P}}(\beta) = \mathbb{E}_{\mathbf{x},y}[l(\beta(\mathbf{x}), y)]$ where the expectation is with respect to the unknown distribution $\mathbb{P}_{\mathbf{x},y}$. Let $\beta^* \in \arg\min_{\beta \in \mathcal{B}_0} l_{\mathbb{P}}(\beta)$ be defined as the 'best-in-class' model with respect to our class $\mathcal{B}_0$. Note that we cannot calculate β^* as we do not have the distribution.

Our set construction method takes into account two things: (i) how the solution β^{Alg} of empirical risk minimization compares with β^* (coming from statistical learning theory), and (ii) how much of the mass of $\mathbb{P}_{\mathbf{x},y}$ concentrates around $\beta^*(\mathbf{x})$ (coming from **Assumption A** described below).

It is always true that there exists a set E and a scalar $\delta_e \geq 0$ such that:

$$\mathbb{P}_{\mathbf{x},y}\left(\mathbf{x}, y : |y - \beta^*(\mathbf{x})| \in E\right) \geq 1 - \delta_e, \tag{5.6}$$

where $E \subseteq \mathcal{Y}$. This is trivially satisfied if $E = \mathcal{Y}$. In this case, δ_e can be set to 0. Ideally, we know of a pair (E, δ_e) where δ_e is still small and where E is not too large; if E were very large, the uncertainty set would be too conservative. We formalize the assumption that we will use to define $\mathcal{U}$ as follows:

Assumption A: We know a pair (E, δ_e) such that Equation (5.6) holds.

We can intuitively think of decomposing $\mathbf{u}$ in Equation (5.2) to capture model uncertainty and residual uncertainty as follows. Let $\mathbf{u}_\beta$ be the part of $\mathbf{u}$ that is derived from a statistical model β. Thus, given $\{\tilde{\mathbf{x}}^j\}_{j=1}^m$, $\mathbf{u}_\beta := [\beta(\tilde{\mathbf{x}}^1) \cdots \beta(\tilde{\mathbf{x}}^m)]^T$. Let the remaining part of $\mathbf{u}$, denoted by $\mathbf{u}_{-\beta}$, be equal to a vector of corresponding model residuals. Thus, $\mathbf{u} = \mathbf{u}_\beta + \mathbf{u}_{-\beta}$.

Let $\mathcal{B}$ represent a set of "good" prediction models. Let $\mathcal{U}$ be equal to $\mathcal{U}_{\mathcal{B}} + \mathcal{U}_{-\mathcal{B}}$ such that $\mathbf{u}_{\beta} \in \mathcal{U}_{\mathcal{B}}$ and $\mathbf{u}_{-\beta} \in \mathcal{U}_{-\mathcal{B}}$. Here, $\mathcal{U}_{\mathcal{B}}$ corresponds to $\mathcal{B}$ in the following way: $\mathcal{U}_{\mathcal{B}} := \{\mathbf{u}_{\beta} : \beta \in \mathcal{B}\}$. On the other hand, $\mathcal{U}_{-\mathcal{B}}$ corresponds to a set that captures the support of most model residuals. Formally,

$$\mathcal{U} = \Pi_{j=1}^{m} \left[\inf\{\beta(\tilde{\mathbf{x}}^j) : \beta \in \mathcal{B}\} - E, \sup\{\beta(\tilde{\mathbf{x}}^j) : \beta \in \mathcal{B}\} + E\right]. \qquad (5.7)$$

An illustration in one dimension, when the set of "good" models has two members, is shown in Figure 5-2(c). If we know the 'best-in-class' model β^*, then $\mathcal{U}_{\mathcal{B}}$ can be a singleton set just containing β^*. Since we do not know β^*, we adapt **Step 1** of the general recipe to construct $\mathcal{U}$ using $\mathcal{U}_{\mathcal{B}}$ and $\mathcal{U}_{-\mathcal{B}}$ as follows:

(a) Define $\mathcal{B}$ using $S = \{(\mathbf{x}^i, y^i)\}_{i=1}^{n}$. Our sets will be of the form (discussed further in Section 5.6) $\mathcal{B} = \{\beta : g(\beta) \leq g(\beta^{Alg}) + c\}$, where g is some function, β^{Alg} is a specific model and c is a parameter. These quantities will depend on the learning algorithm and $\{(\mathbf{x}^i, y^i)\}_{i=1}^{n}$.

(b) Define $\mathcal{U}_{\mathcal{B}}$ and $\mathcal{U}_{-\mathcal{B}}$: Recall that $\mathcal{U}_{\mathcal{B}} := \{\mathbf{u}_{\beta} : \beta \in \mathcal{B}\}$ where $\mathbf{u}_{\beta} = [\beta(\tilde{\mathbf{x}}^1) \cdots \beta(\tilde{\mathbf{x}}^m)]^T$. $\mathcal{U}_{-\mathcal{B}}$ is defined using assumption **Assumption A** such that it captures the support of the model error residuals (more details are in Section 5.6). $\mathcal{U}$ is then $\mathcal{U}_{\mathcal{B}} + \mathcal{U}_{-\mathcal{B}}$.

The quality of the robust solution of Equation (5.2) depends on the set E. For a less conservative solution, we want set E to be as small as possible. The probabilistic guarantee on the robust solution that we derive in Section 5.6 depends on δ_e. For a better guarantee, we need δ_e to be as close as possible to 0. If our model class $\mathcal{B}_0$ is very complex and able to closely capture most y values, this could reduce the size of set E.

Note that if $\mathcal{B}$ does not contain good models, $\mathcal{U}_{-\mathcal{B}}$ will necessarily be large, our bound on robustness will be loose, and the robust solution thus obtained will be too conservative.

Using two sets of "good" prediction models:

When our prediction problem is a regression task, we can make a different (and often weaker) assumption than **Assumption A** using quantile regression. We will construct uncertainty set $\mathcal{U}$ in a different way.

Recall the definition for the conditional quantile function $\mu^\tau(\mathbf{x})$ and the empirical procedure to estimate it, outlined in Section 5.3.1. Let $\beta^{\tau,*} \in \arg\min_{\beta\in\mathcal{B}_0} l^\tau_{\mathbf{P}}(\beta)$ be the 'best-in-class' conditional quantile function for any given τ. It is always true that there exists a set $E^\tau \subseteq \mathcal{Y}$ and a scalar $\delta^\tau_e \geq 0$ such that:

$$\mathbb{P}_{\mathbf{x}}(\mathbf{x} : |\mu^\tau(\mathbf{x}) - \beta^{\tau,*}(\mathbf{x})| \in E^\tau) \geq 1 - \delta^\tau_e. \tag{5.8}$$

The way we will construct $\mathcal{U}$ below will be such that the quality of the robust solution $\boldsymbol{\pi}^*$ of Equation (5.2) depends on the set E^τ. For a less conservative solution, we want E^τ to be as small as possible. The probabilistic guarantee that we derive in Section 5.7 on $\boldsymbol{\pi}^*$ will depend on δ^τ_e. For a better guarantee, δ^τ_e needs to be as close as possible to 0. If $\mathcal{B}_0$ is sufficiently rich, E^τ can be small or even empty (which is the case when $\mu^\tau \in \mathcal{B}_0$). Thus, similar to **Assumption A** in Section 5.3.2, we make the following assumption:

Assumption B: Given a value of τ, we know a pair (E^τ, δ^τ_e) such that Equation (5.8) holds.

Let $\mathcal{B}^{\delta_p}$ be the set of "good" conditional quantile functions when $\tau = \delta_p$ and let $\mathcal{B}^{\delta_q}$ be the set of "good" conditional quantile functions when $\tau = \delta_q$. By "good" we mean that all these quantile functions have their quantile estimation performance close to the best we can obtain from $\mathcal{B}_0$ using quantile regression. A precise definition for $\mathcal{B}^\tau$ will be given in Equation (5.15) below. We can then construct $\mathcal{U}$ as:

$$\begin{aligned}\mathcal{U} = \Pi^m_{j=1}\Big[&\inf\{\beta(\tilde{\mathbf{x}}^j) : \beta \in \mathcal{B}^{\delta_p} \cup \mathcal{B}^{\delta_q}\} - \sup E^{\delta_p} \cup E^{\delta_q},\\ &\sup\{\beta(\tilde{\mathbf{x}}^j) : \beta \in \mathcal{B}^{\delta_p} \cup \mathcal{B}^{\delta_q}\} + \sup E^{\delta_p} \cup E^{\delta_q}\Big].\end{aligned} \tag{5.9}$$

The definition of the j^{th} interval involves two sets. The first set, $\{\beta(\tilde{\mathbf{x}}^j) : \beta \in \mathcal{B}^{\delta_p} \cup \mathcal{B}^{\delta_q}\}$ contains all the predictions by models in both $\mathcal{B}^{\delta_q}$ and $\mathcal{B}^{\delta_q}$ on the feature vector

$\tilde{\mathbf{x}}^j$. The second set, $E^{\delta_p} \cup E^{\delta_q}$, contains all deviations between the true conditional quantiles and the 'best-in-class' conditional quantiles at both values of τ. Thus, the smallest value of the predicted δ_p conditional quantiles and δ_q conditional quantiles in the first set, in conjunction with the largest deviation captured by the second set, is used to define the lower limit of the interval. The upper limit of the interval is defined in a similar way by taking the largest predicted quantile from the first set and adding the largest deviation captured by the second set. An illustration in one dimension, when each of the two sets of "good" models has two members, is shown in Figure 5-2(d).

Given $\mathcal{U}$, we can solve Equation (5.2) for $\boldsymbol{\pi}^*$ using **Step 2**. The following is a summary of the way to construct $\mathcal{U}$ in **Step 1**:

(a) Define $\mathcal{B}^{\delta_p}$ and $\mathcal{B}^{\delta_q}$ using $\{(\mathbf{x}^i, y^i)\}_{i=1}^n$. We will propose procedures for designing $\mathcal{B}^{\delta_p}$ and $\mathcal{B}^{\delta_q}$ using learning theory results in Section 5.7. Our sets will be of the form $\mathcal{B}^{\tau} = \{\beta : g(\beta) \leq g(\beta^{Alg,\tau}) + c\}$ for $\tau = \delta_p, \delta_q$, where g is some function, $\beta^{Alg,\tau}$ is a specific conditional quantile model depending on τ and c is a parameter. These quantities will depend on the pinball loss function and $\{(\mathbf{x}^i, y^i)\}_{i=1}^n$.

(b) Define $\mathcal{U}$: Using the above sets and the property of quantile error residuals as in Equation (5.8), and by **Assumption B**, we can construct $\mathcal{U}$ as shown in Equation (5.9).

In the next few sections, we provide probabilistic guarantees on the feasibility of the robust optimal solutions obtained by using uncertainty sets from each of the four methods we proposed.

5.4 Robustness guarantee using general prediction functions

Consider the setting described in Section 5.3, where we have a class of general set functions $\mathcal{I}$. Let $S := \{(\mathbf{x}^i, y^i)\}_{i=1}^n$ be the training data which are independent and identically distributed. Let algorithm A represent a generic learning procedure. That is, it takes S as an input and outputs I^{Alg}. Since I^{Alg} is a function of sample S, we will show that the unknown $\tilde{y}^j$ belong to the interval $I^{\mathrm{Alg}}(\tilde{\mathbf{x}}^j)$ with high probability over S as long as the set of functions $\mathcal{I}$ from which I^{Alg} is picked is "simple". Note that we do not assume anything about the source distribution.

In order to state our result, we will define the following quantity known as the empirical Rademacher average [Bartlett et al., 2002]. For a set $\mathcal{F}$ of functions, the *empirical Rademacher average* is defined with respect to a given random sample $S' = \{z^i\}_{i=1}^n$ as $\mathcal{R}_{S'}(\mathcal{F}) = \mathbb{E}_{\sigma^1,...,\sigma^n}\left[\frac{1}{n}\sup_{f\in\mathcal{F}}|\sum_{i=1}^n \sigma^i f(z^i)|\right]$ where for each $i = 1,..,n$, $\sigma^i = \pm 1$ with equal probability. The *Rademacher average* is defined to be the expectation of the empirical Rademacher average over the random sample S: $\mathcal{R}(\mathcal{H}) = \mathbb{E}_{z^1,...,z^n}[\mathcal{R}_S(\mathcal{H})]$. The interpretation of the Rademacher average is that it measures the ability of function class $\mathcal{F}$ to fit noise, coming from the random $\sigma_i's$. If the function class can fit noise well, it is a highly complex class. The Rademacher average is one of many ways to measure the richness of a function class, including covering numbers, fat-shattering dimensions [Bartlett et al., 1996] and the Vapnik-Chervonenkis dimension [Vapnik, 1998].

Theorem 5.4.1. If $\mathcal{U}$ is defined as in Equation (5.4), then with probability at least $1-\delta$ over training sample S, we have robustness guarantee

$$\mathbb{P}_{\{\tilde{\mathbf{x}}^j,\tilde{y}^j\}_{j=1}^m}\left(F(\pi^*,[\tilde{y}^1...\tilde{y}^m]^T)\in\mathcal{K}\right) \geq \left(\left[1-\frac{1}{n}\sum_{i=1}^n 1[y^i\notin I^{\mathrm{Alg}}(\mathbf{x}^i)] - 2\mathcal{R}(l\circ\mathcal{I}) - \sqrt{\frac{\log\frac{1}{\delta}}{2n}}\right]_+\right)^m,$$

where $\epsilon > 0$ is a pre-determined constant, and $\left[\cdot\right]_+$ is shorthand for $\max(0,\cdot)$.

The result is a lower bound on the probability of infeasibility. This bound depends on the performance of the data dependent set function I^{Alg}. If I^{Alg} is such that its performance, measured in terms of $\frac{1}{n}\sum_{i=1}^n 1[y^i \notin I^{\mathrm{Alg}}(\mathbf{x}^i)]$ is good (i.e., lower in value), then the right hand side of the inequality increases, resulting in a higher chance of feasibility. This probability of feasibility also depends on the number of estimates m that enter the decision problem of Equation (5.2). When $n\to\infty$, the

Rademacher term and the square root terms become zero and the probability of feasibility depends on the asymptotic performance of I^{Alg} (which converges to I^*, the 'best-in-class' set function), as desired. The proof is provided in Section 5.8.3.

5.5 Robustness guarantee using conditional quantile functions

Theorem 5.5.1. If $\mathcal{U}$ is defined as in Equation (5.5), then with probability at least $1 - \delta$ over training sample S, we have

$$\mathbb{P}_{\{\tilde{\mathbf{x}}^j,\tilde{y}^j\}_{j=1}^m}\Big(F(\pi^*,[\tilde{y}^1 ... \tilde{y}^m]^T) \in \mathcal{K}\Big) \geq \\ \left(\left[\frac{1}{n}\sum_{i=1}^{n}\Big(r_\epsilon^-(y^i - \beta^{\text{Alg},\delta_q}(\mathbf{x}^i)) - r_\epsilon^+(y^i - \beta^{\text{Alg},\delta_p}(\mathbf{x}^i))\Big) - \frac{8}{\epsilon}\mathcal{R}(\mathcal{B}_0) - 2\sqrt{\frac{\log\frac{2}{\delta}}{2n}}\right]_+\right)^m, \tag{5.10}$$

where $\epsilon > 0$ is a pre-determined constant, $\Big[\cdot\Big]_+$ is shorthand for $\max(0,\cdot)$, $r_\epsilon^-(z) := \min\Big(1, \max\big(0, -\frac{z}{\epsilon}\big)\Big)$ and $r_\epsilon^+(z) := \min\Big(1, \max\big(0, 1 - \frac{z}{\epsilon}\big)\Big)$.

The robustness guarantee is established by replacing $\mathbb{P}_{\mathbf{x},y}(y \leq \beta(\mathbf{x}))$ with the expectation of a related indicator random variable. By majorizing this random variable by random variables defined using functions r_ϵ^- and r_ϵ^+, we were able to use the machinery of Rademacher concentration results. The proof is provided in Section 5.8.2.

5.6 Robustness guarantee using a single set of good models

Here we consider the third method prescribed in Section 5.3.2. Let $\beta^{Alg} \in \mathcal{B}_0$ be the model output by the empirical risk minimization procedure. Let empirical risk $l_S(\beta) = \frac{1}{n}\sum_{i=1}^n l(\beta(\mathbf{x}^i), y^i)$ be a function of our sample S. Let A produce β^{Alg} according to $\beta^{Alg} \in \arg\min_{\beta \in \mathcal{B}_0} l_S(\beta)$. That is, the algorithm A is minimizing the empirical loss. We define $\mathcal{B}$ using the empirical Rademacher average, defined in Section 5.4, as follows:

$$\mathcal{B} := \left\{ \beta \in \mathcal{B}_0 : l_S(\beta) \leq l_S(\beta^{Alg}) + 2\mathcal{R}_S(l \circ \mathcal{B}_0) + 4M\sqrt{\frac{\log \frac{3}{\delta}}{2n}} \right\}, \tag{5.11}$$

where M is a bound on the range of the loss function l, and δ is pre-specified and parameterizes the probabilistic guarantee on the robust optimal solution. $\mathcal{R}_S(l \circ \mathcal{B}_0)$ is the empirical Rademacher average of the function class $l \circ \mathcal{B}_0 := \{l(\beta(\cdot), \cdot) : \beta \in \mathcal{B}_0\}$.

We define $\mathcal{U}_{-\mathcal{B}} := E^m$ (m copies of E) where E satisfies Equation (5.6) for a given δ_e and m is the number of predictions (equal to the length of the vector $\mathbf{u}_\beta$). Intuitively, $\mathcal{U}_{-\mathcal{B}}$ is capturing the support of prediction errors if we knew the 'best-in-class' model β^*. Recall that these definitions of $\mathcal{U}_\mathcal{B}$ and $\mathcal{U}_{-\mathcal{B}}$ lead to the set $\mathcal{U}$ in Equation (5.7).

Theorem 5.6.1. If $\mathcal{U}$ is defined as in Equation (5.7), then the following hold:

1. With probability at least $1 - \delta$, $\beta^* \in \mathcal{B}$.

2. Robust optimal solution $\boldsymbol{\pi}^*$ of Equation (5.2) is feasible for $\{(\tilde{\mathbf{x}}^j, \tilde{y}^j)\}_{j=1}^m$ with probability at least $(1-\delta)(1-\delta_e)^m$ over $\{(\tilde{\mathbf{x}}^j, \tilde{y}^j)\}_{j=1}^m$ and S. That is,

$$\mathbb{P}_{S, \{(\tilde{\mathbf{x}}^j, \tilde{y}^j)\}_{j=1}^m} \left(F(\boldsymbol{\pi}^*, [\tilde{y}^1 \ \dots \ \tilde{y}^m]^T) \in \mathcal{K} \right) \geq (1-\delta)(1-\delta_e)^m.$$

The above theorem holds for any bounded loss function l. It guarantees that π^* will be robust to parameter $\mathbf{u}$ with components $\mathbf{u}_\beta = [\beta^*(\tilde{\mathbf{x}}^1) \dots \beta^*(\tilde{\mathbf{x}}^j) \dots \beta^*(\tilde{\mathbf{x}}^m)]^T$ and $\mathbf{u}_{-\beta} = [\tilde{y}^1 \dots \tilde{y}^j \dots \tilde{y}^m]^T - [\beta^*(\tilde{\mathbf{x}}^1) \dots \beta^*(\tilde{\mathbf{x}}^j) \dots \beta^*(\tilde{\mathbf{x}}^m)]^T$ because the sum of these components is equal to $[\tilde{y}^1 \ \dots \ \tilde{y}^m]^T$.

We insure against most possible realizations of $\{\tilde{y}^j\}_{j=1}^m$ in a particular way: by first ensuring β^* belongs to $\mathcal{B}$ with high probability (see Theorem 5.6.1 part (1)) and then ensuring that the random errors $\tilde{y}^j - \beta^*(\tilde{\mathbf{x}}^j)$ are in $\mathcal{U}_{-\mathcal{B}}$ also with high probability. Thus the $\{\tilde{y}^j\}_{j=1}^m$ belong to the set $\mathcal{U}_\mathcal{B} + \mathcal{U}_{-\mathcal{B}}$ with high probability.

This theorem also tells us how the choice of $\mathcal{B}_0$ affects the size of our uncertainty set precursor $\mathcal{B}$. Interestingly enough, if we work with a (possibly infinite) set of predictive models $\mathcal{B}_0$ such that its empirical Rademacher average $\mathcal{R}_S(l \circ \mathcal{B}_0)$ scales as $O(n^{-\frac{1}{2}})$, then we have similar quantitative dependence on n compared to that of confidence-interval based approaches (that make explicit distributional assumptions - see Section 5.6.3 - whereas we do not need to make such assumptions). In fact, for many well studied model classes the scaling of the empirical Rademacher average is indeed $O(n^{-\frac{1}{2}})$ which we will review shortly.

One of the advantages of defining uncertainty set precursor $\mathcal{B}$ in the way we proposed is that it directly links the uncertainty in decision making to the loss function $l(\beta(\mathrm{x}), y)$ and sample S of the machine learning step. One advantage of using the empirical Rademacher average in defining $\mathcal{B}$ is that it makes use of the data sample S in its definition, and can reflect the properties of the particular unknown distribution $\mathbb{P}_{\mathbf{x},y}$ of the data source.

5.6.1 Robustness guarantees when the hypothesis set $\mathcal{B}_0$ is finite:

When $\mathcal{B}_0$ consists of a finite number of models, we can define $\mathcal{B}$ without using the notion of Rademacher averages. Let $|\mathcal{B}_0|$ represent the size of the set $\mathcal{B}_0$. Then we can define the set of good models as:

$$\mathcal{B} := \left\{ \beta \in \mathcal{B}_0 : l_S(\beta) \leq l_S(\beta^{Alg}) + M\sqrt{\frac{\log|\mathcal{B}_0| + \log\frac{2}{\delta}}{2n}} + M\sqrt{\frac{\log\frac{2}{\delta}}{2n}} \right\}, \tag{5.12}$$

where $n, \delta, M, l_S(\cdot)$ and β^{Alg} have the same definitions as before.

Theorem 5.6.2. For finite $\mathcal{B}_0$, the conclusion of Theorem 5.6.1 holds if $\mathcal{U}$ in Equation (5.7) is defined using $\mathcal{B}$ described in Equation (5.12).

5.6.2 Constructing $\mathcal{U}$ using PAC-Bayes theory:

If the learning step is a classification task, we can also define $\mathcal{B}$ using the PAC-Bayes framework of McAllester [1999], where PAC means "probably approximately correct". This framework does not seek a single empirically good classifier β^{Alg} and instead finds a good "posterior" distribution Q over the hypothesis set $\mathcal{B}_0$. The corresponding theory provides a probabilistic guarantee on the performance of the classifiers that holds uniformly over all posterior distributions within a class of distributions. The framework then picks a Q using data sample S so that a Q-weighted deterministic classifier (or a Q-based randomized classifier) has the optimal probabilistic guarantee.

Consider the Q-based (randomized) Gibbs classifier G_Q, which makes each prediction by choosing a classifier from $\mathcal{B}_0$ according to Q. Let the Q-based Gibbs classifier have the following definitions of risks: (a) expected risk $R(G_Q) := \mathbb{E}_{\beta \in Q}[l_{\mathbf{P}}(\beta)]$, and (b) empirical risk $R_S(G_Q) := \mathbb{E}_{\beta \in Q}[l_S(\beta)]$ where $l_{\mathbf{P}}(\beta)$ and $l_S(\beta)$ are the same as in Section 5.6. The PAC-Bayes framework guarantees that for all Q, $R(G_Q)$ is bounded by $R_S(G_Q)$ and a term which captures the deviation of Q from a pre-specified 'prior' distribution P over $\mathcal{B}_0$ as follows:

Theorem 5.6.3. Germain et al. [2009, Theorem 2.1]: Let $l(\beta(\mathbf{x}), y) := \mathbf{1}[\beta(\mathbf{x}) \neq y]$. For any $\mathbb{P}_{\mathbf{x},y}$, any $\mathcal{B}_0$, any prior P on $\mathcal{B}_0$, any $\delta \in (0, 1]$ and any convex function $\mathcal{D} : [0,1]^2 \to \mathbb{R}$, we have

$$\mathbb{P}_S\left(\forall Q \text{ on } \mathcal{B}_0 : \mathcal{D}(R_S(G_Q), R(G_Q)) \le \frac{1}{n}\left[KL(Q||P) + \log\left(\frac{1}{\delta}\mathbb{E}_S\mathbb{E}_{\beta\sim P}e^{m\mathcal{D}(l_S(\beta), l_{\mathbf{P}}(\beta))}\right)\right]\right) \ge 1 - \delta, \tag{5.13}$$

where $KL(Q||P) := \mathbb{E}_{\beta\sim Q}[\log \frac{Q(\beta)}{P(\beta)}]$.

As shown by Germain et al. [2009], for a certain choice of the metric $\mathcal{D}$ the above theorem gives a bound on $R(G_Q)$ that is proportional to $CnR_S(G_Q) + KL(Q||P)$ where C is a pre-specified constant. We can minimize this quantity to get an optimal distribution Q^{Alg} with a closed form expression: $Q^{\text{Alg}}(\beta) = \frac{1}{Z}P(\beta)e^{-Cnl_S(\beta)}$ where Z

is a normalizing constant.

The set of good models $\mathcal{B}$, for the model uncertainty set $\mathcal{U}_{\mathcal{B}}$, can be defined by setting Q^{Alg} to be bigger than a threshold, leading to:

$$\mathcal{B} = \left\{ \beta \in \mathcal{B}_0 : l_S(\beta) \leq \frac{\log P(\beta) - \alpha}{nC} \right\},$$

where $\alpha > 0$ is a fixed constant, $P(\beta)$ is the prior probability density of model β, and C is a constant that appears in the objective when we solve for Q^{Alg}. Intuitively, the set $\mathcal{B}$ includes all models such that their empirical error is bounded in a way that considers their scaled log prior density values. By our construction, if $\beta \in \mathcal{B}$, then $Q^{\text{Alg}}(\beta)$ is greater than the threshold $\frac{e^{\alpha}}{Z}$. There is no notion of a 'best-in-class' model β^* in the PAC-Bayes setting and thus we do not have a guarantee similar to Theorem 5.6.1. Nonetheless, $\mathcal{B}$ is data driven and captures those models that have a high posterior density in $\mathcal{B}_0$. $\mathcal{U}_{\mathcal{B}}$ and $\mathcal{U}_{-\mathcal{B}}$ are defined using $\mathcal{B}$ and Equation (5.6) in the same way as before and used to obtain π^*.

5.6.3 Contrasting this method with that of Goldfarb and Iyengar [2003]:

Goldfarb and Iyengar [2003] assume distributional properties on $\{(\mathbf{x}^i, y^i)\}_{i=1}^n$ (**Assumptions GI1**) in addition to assuming a functional form for $\beta(\mathbf{x})$ (**Assumption GI2**) while working with robust portfolio selection problems. In particular, let $y = \beta(\mathbf{x}) + \epsilon$, where $\beta(\mathbf{x}) = \boldsymbol{\beta}^T\mathbf{x}$ is the functional form of the model. Let us assume that $\mathbf{x}^i \in \mathcal{X} \subseteq \mathbf{R}^d$ are chosen by the experimenter and are not random. The only source of randomness is through ϵ which is independent from example to example and is assumed to be distributed according to $\mathcal{N}(0, \sigma^2)$ with variance σ^2 known. Then an estimator of $\boldsymbol{\beta}^*$ (the 'best-in-class' model) is given by: $\boldsymbol{\beta}^{\text{Alg}} = (X^TX)^{-1}X^TY$, where X is a matrix with n rows, one for each $\mathbf{x}^i$ and Y is an $n \times 1$ vector with the ith element being y^i. Here assume that X^TX is invertible. Substituting $Y = X\boldsymbol{\beta}^* + \epsilon$ in the expression for $\boldsymbol{\beta}^{\text{Alg}}$ gives us: $\boldsymbol{\beta}^{\text{Alg}} - \boldsymbol{\beta}^* = (X^TX)^{-1}X^T\epsilon$, which is then distributed as $\mathcal{N}(0, \sigma^2(X^TX)^{-1})$. Thus, the real-valued function $g(\boldsymbol{\beta}^*, S) := \frac{1}{\sigma^2}(\boldsymbol{\beta}^{\text{Alg}} - \boldsymbol{\beta}^*)^T(X^TX)(\boldsymbol{\beta}^{\text{Alg}} - \boldsymbol{\beta}^*)$ is a χ_d^2

distributed random variable. Because of this, we can find a range such that with high probability the χ^2_d distributed random variable $g(\boldsymbol{\beta}^*, S)$ belongs to it. We can adapt this approach to our notation by choosing $\mathcal{B}$ based on this interval, giving us an ellipsoid centered at $\boldsymbol{\beta}^{\text{Alg}}$: $\mathcal{B} = \left\{\boldsymbol{\beta} : \frac{1}{\sigma^2}(\boldsymbol{\beta}^{\text{Alg}} - \boldsymbol{\beta})(X^T X)(\boldsymbol{\beta}^{\text{Alg}} - \boldsymbol{\beta}) \leq c\right\}$, where c is a constant that determines how much of the probability mass of χ^2_d is within the set $\mathcal{B}$.

Set $\mathcal{U}_{-\mathcal{B}}$ can be defined using our assumption about the model residuals: $\epsilon = (y - \boldsymbol{\beta}^T x) \sim \mathcal{N}(0, \sigma^2)$. In particular, using Equation (5.6), we can obtain interval $E = [-e, e]$ for any desired value of δ_e by solving the equation: $\int_{-e}^{e} \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{s}{2\sigma^2}} ds = 1 - \delta_e$.

Using $\mathcal{B}$ (equivalently, $\mathcal{U}_{\mathcal{B}}$) and $\mathcal{U}_{-\mathcal{B}}$ as defined above in the robust problem of Equation (5.2) gives us a guarantee on the robustness of π^* to future realizations of y if **Assumptions GI1** and **Assumption GI2** hold. If noise variance σ^2 is unknown, regression theory provides the following fix: we obtain an unbiased estimator of σ^2 given by $s^2 = \frac{\|Y - X\boldsymbol{\beta}^{\text{Alg}}\|_2^2}{n-d}$. The resulting scaled random variable $\frac{1}{ds^2}(\boldsymbol{\beta}^{\text{Alg}} - \boldsymbol{\beta})(X^T X)(\boldsymbol{\beta}^{\text{Alg}} - \boldsymbol{\beta})$ is F-distributed with d degrees of freedom in the numerator and $n - d$ degrees of freedom in the denominator [Anderson, 1958]. A set of good models $\mathcal{B}$ can be defined in the same way as before. The constant c now determines how much of the probability mass of an $F_{d,n-d}$-distributed random variable is within $\mathcal{B}$.

Note that both **Assumptions GI1** and **Assumption GI2** (or their variations for similar models) are heavily needed to justify these constructions. Contrast this with the setting of Section 5.6 where much weaker assumptions were made and the setting of Section 5.4, where the only assumption made is that the data are drawn i.i.d from some distribution. Because our assumptions are much weaker, our result applies to many different loss functions and lends itself naturally to many different machine learning approaches.

Evaluating the empirical Rademacher average: In the expression for $\mathcal{B}$ in Equation (5.11), it may sometimes be difficult to compute the value of $\mathcal{R}_S(l \circ \mathcal{B}_0)$ efficiently. In these cases, we have two options. The first one involves finding upper bounds on

$\mathcal{R}_S(l \circ \mathcal{B}_0)$. This can be tricky as $\mathcal{R}_S$ depends on the data. The second one involves defining $\mathcal{B}$ directly in terms of the *Rademacher average* $\mathcal{R}(l \circ \mathcal{B}_0)$:

$$\mathcal{B} := \left\{ \beta \in \mathcal{B}_0 : l_S(\beta) \leq l_S(\beta^{Alg}) + 2\mathcal{R}(l \circ \mathcal{B}_0) + 3M\sqrt{\frac{\log \frac{2}{\delta}}{2n}} \right\}. \qquad (5.14)$$

It can be shown that the optimal robust solution obtained using the set in (5.14) enjoys a guarantee similar to the solution obtained using the set in (5.11) with different constants. We can make use of the various relationships in Theorem 12 of Bartlett and Mendelson [2002] to upper bound $\mathcal{R}(l \circ \mathcal{B}_0)$ or $\mathcal{R}_S(l \circ \mathcal{B}_0)$ analytically. The following are some examples:

- For linear function classes with squared loss as the loss function, we have: $\mathcal{R}(\mathcal{B}_0) \leq \frac{X_b B_b}{\sqrt{n}}$, and $\mathcal{R}(l \circ \mathcal{B}_0) \leq 8X_b B_b \frac{X_b B_b}{\sqrt{n}}$. where the latter inequality uses Corollary 3.17 in Ledoux and Talagrand [1991] that relates $\mathcal{R}(l \circ \mathcal{B}_0)$ and $\mathcal{R}(\mathcal{B}_0)$. That is, when the loss function $l(\beta(\mathbf{x}), y)$ is $\mathcal{L}$-Lipschitz we have: $\mathcal{R}(l \circ \mathcal{B}_0) \leq 2\mathcal{L} \cdot \mathcal{R}(\mathcal{B}_0)$. For the squared loss function, $\mathcal{L} = 4X_b B_b$ if $\forall \mathbf{x} \in \mathcal{X}, \|\mathbf{x}\|_2 \leq X_b$ and $\forall \beta \in \mathcal{B}_0, \|\beta\|_2 \leq B_b$. Note that this bound does not depend on data sample S.

- For kernel based function classes with Lipschitz loss functions, $\mathcal{B}_0$ can be written as:

$$\mathcal{B}_0 = \left\{ x \mapsto \sum_{i=1}^{n} \alpha^i k(x, x^i) : n \in \mathbb{N}, x \in \mathcal{X}, \sum_{i,j} \alpha^i \alpha^j K(x^i, x^j) \leq B_b \right\},$$

  where $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a bounded kernel (k is called a kernel if an $n \times n$ Gram matrix K with entries $(K)_{i,j} = k(x^i, x^j)$ is positive semi-definite). This function class is used in Support Vector Machines (SVMs) [e.g., see Cristianini and Shawe-Taylor, 2000] where the loss function is the hinge-loss. The following bound [see Bartlett and Mendelson, 2002, Lemma 22] applies when the loss function is $\mathcal{L}$-Lipschitz (as is the hinge loss):

$$\mathcal{R}_S(\mathcal{B}_0) \leq \frac{B_b}{n}\sqrt{\sum_{i=1}^{n} k(x^i, x^j)}, \quad \text{and} \quad \mathcal{R}_S(l \circ \mathcal{B}_0) \leq 2\mathcal{L}\frac{B_b}{n}\sqrt{\sum_{i=1}^{n} k(x^i, x^j)}.$$

This upper bound reduces to the previous case (linear function class and squared loss) when we choose the appropriate kernel and loss function. In particular, using the dot product kernel $k(x^i, x^j) = (x^i)^T x^j$ we get:

$$\mathcal{R}_S(l \circ \mathcal{B}_0) \leq 2\mathcal{L}\frac{B_b}{n}\sqrt{\sum_{i=1}^{n} k(x^i, x^j)} = 2\mathcal{L}\frac{B_b}{n}\sqrt{\sum_{i=1}^{n} (x^i)^T x^j} \leq 2\mathcal{L}\frac{B_b}{n}\sqrt{nX_b^2} = 8X_b B_b \frac{X_b B_b}{\sqrt{n}}.$$

5.7 Robustness guarantee using sets of good conditional quantile models

Consider the fourth method prescribed in Section 5.3.2. Let us define $\mathcal{B}^{\delta_p}$ and $\mathcal{B}^{\delta_q}$ using $\{(\mathbf{x}^i, y^i)\}_{i=1}^n$ in a very similar way to defining $\mathcal{B}$ in Equation (5.11). Let the empirical risk minimization procedure using the pinball loss output a conditional quantile model $\beta^{Alg,\tau}$, given sample $S = \{(\mathbf{x}^i, y^i)\}_{i=1}^n$ of size n and parameter τ. That is, let $l_S^\tau(\beta) = \frac{1}{n}\sum_{i=1}^n l^\tau(\beta(\mathbf{x}^i), y^i)$ and $\beta^{Alg,\tau} \in \arg\min_{\beta \in \mathcal{B}_0} l_S^\tau(\beta)$. The following definition of $\mathcal{B}^\tau$ gives us the two sets when $\tau = \delta_p$ and $\tau = \delta_q$:

$$\mathcal{B}^\tau := \left\{\beta \in \mathcal{B}_0 : l_S^\tau(\beta) \leq l_S^\tau(\beta^{Alg,\tau}) + 2\mathcal{R}_S(l^\tau \circ \mathcal{B}_0) + 4M\sqrt{\frac{\log \frac{3}{\delta}}{2n}}\right\}, \qquad (5.15)$$

where M is a bound on the range of the loss function l^τ, δ is a pre-specified constant and $\mathcal{R}_S(l^\tau \circ \mathcal{B}_0)$ is the empirical Rademacher average of the function class $l^\tau \circ \mathcal{B}_0 := \{\beta \mapsto l^\tau(\beta(\cdot), \cdot) : \beta \in \mathcal{B}_0\}$. The guarantee on the robust optimal solution of Equation (5.2) is given by the following theorem.

Theorem 5.7.1. If $\mathcal{U}$ is defined as described in Equation (5.9), using sets $\mathcal{B}^{\delta_p}, \mathcal{B}^{\delta_q}$ defined in Equation (5.15) and set E in Equation (5.8) along with **Assumption B**, then the following hold:

1. With probability at least $1 - \delta$, $\beta^{\tau,*} \in \mathcal{B}^\tau$ for $\tau = \delta_p$ and $\tau = \delta_q$ individually.

2. Robust optimal solution $\boldsymbol{\pi}^*$ of Equation (5.2) is feasible for $\{(\tilde{\mathbf{x}}^j, \tilde{y}^j)\}_{j=1}^m$ with probability at least $(1-\delta)\left[(1 - \delta_e^{\delta_p})^m + (1 - \delta_e^{\delta_q})^m\right] + (\delta_q - \delta_p)^m - 2$ over $\{(\tilde{\mathbf{x}}^j, \tilde{y}^j)\}_{j=1}^m$

and S. That is,

$$\mathbb{P}_{S,\{(\tilde{\mathbf{x}}^j,\tilde{y}^j)\}_{j=1}^m}\left(F(\pi^*,[\tilde{y}^1...\tilde{y}^m]^T)\in\mathcal{K}\right)\geq(1-\delta)\left[(1-\delta_e^{\delta_p})^m+(1-\delta_e^{\delta_q})^m\right]+(\delta_q-\delta_p)^m-2.$$

In the theorem, the guarantee follows from designing $\mathcal{U}$ such that the predictions made by the 'best-in-class' conditional quantile functions $\beta^{\delta_p,*}$, $\beta^{\delta_q,*}$ and their residuals are captured in each interval defining $\mathcal{U}$. This ensures that the realization $[\tilde{y}^1...\tilde{y}^m]^T\in\mathcal{U}$ with high probability.

The guarantees in Sections 5.6 and this section do not assume anything about the form of the source distribution. These bounds do what learning theory is designed to do [Bousquet, 2003], which is provide insight into the important quantities for learning and how they scale. More importantly, here they provide insight beyond prediction, specifically into robustness for decision making. We generally do not use learning theoretic bounds directly in practice (e.g., SVMs do not minimize the generalization bounds that motivated their derivation). To translate our results in practice, we suggest using our workflow to construct the uncertainty sets as in Equations (5.11), (5.12) or (5.14), replacing the Rademacher average term with an appropriate choice of parameters. A practitioner can also perform a type of sensitivity analysis for our approach by varying the size of the uncertainty sets and assessing the corresponding results.

5.7.1 Insights and Comparison of Main Results

Before we move onto the proofs, we recap the main results. Theorem 5.4.1 provides a very general results that pertains to any algorithm. Intuitively, it states that as long as the algorithm's result is robust to most of the training examples, and as long as the algorithm can only produce simple functions, it will likely be robust to all points in the test set. This is true for any unknown distribution of data, with no assumptions on the distribution.

Theorem 5.4.1 does not provide any insights on how to construct an algorithm for data-driven robust optimization, since it holds for any algorithm. Theorem 5.5.1, on the other hand, provides a result that holds for quantile regression methods. Here we use an algorithm that produces an estimate for a lower quantile and an estimate for a higher quantile, and chooses the policy to be robust to all points between these quantile estimates. The result applies to any method for producing such estimates. It states that, for this choice of policy, the solution will be robust to all points on the test set with high probability. The bound will be tighter when the class of quantile estimation functions produced by the algorithm is simpler. Theorem 5.5.1 is close to being a special case of Theorem 5.4.1. Theorem 5.5.1's loss function is similar to a special case of Theorem 5.4.1's, and the complexity term differs only through a Lipschitz constant of the loss function which is explicitly taken into account in Theorem 5.5.1 but not in Theorem 5.4.1.

Theorems 5.6.1 and 5.7.1 rely on mild probabilistic assumptions that can make the bounds tighter. These theorems consider the full set of "good" models, that is, models with small loss on the training set, and expand outwards to include more points into the uncertainty set; thus these theorems take into account both the behavior on the training set and the assumed behavior on the full distribution of data.

For the assumption underlying Theorems 5.6.1 and 5.7.1, recall that δ_e is the probability that the tails of the distribution are within E of the true mean or quantile estimates. There is a tradeoff in assumptions between E and δ_e, in the sense that the policy needs to be robust to a larger uncertainty set if E is large; larger E leads to conservative policy choices. At the same time, if E is larger, our assumption that E includes the tails of the distribution should be stronger, leading to smaller δ_e. When δ_e is smaller, the probabilistic guarantee on robustness is also stronger.

Theorem 5.6.1's result holds for any algorithm that produces estimates of centrality for y given x (e.g., mean or median). Theorem 5.7.1's result holds for any algorithm that produces quantile estimates. We believe that the uncertainty set construction used for Theorem 5.7.1 is the most natural ones to use, regardless of whether the assumptions relating δ_e and E hold precisely. To recap, this is where we compute the highest estimate of the upper quantile from all good models, compute the lowest estimate of the lower quantile from all good models, and expand outwards, to produce the uncertainty set.

5.8 Proofs

Before we proceed with the proofs of guarantees for the four methods in Sections 5.4-5.7, we state an intermediate result we will make use of in all four proofs. This result gives a uniform probabilistic guarantee on the deviation between empirical loss and expected loss of prediction models in terms of the Rademacher average. It holds for any set of models $\mathcal{F}$ and a bounded loss function l.

Lemma 5.8.1. With probability at least $1-\delta$ over sample S,

$$\max_{f\in\mathcal{F}} |l_{\mathbb{P}}(f) - l_S(f)| \leq 2\mathcal{R}(l\circ\mathcal{F}) + M\sqrt{\frac{\log\frac{1}{\delta}}{2n}}.$$

Proof. Here, $\max_{f\in\mathcal{F}}|l_{\mathbb{P}}(f) - l_S(f)|$ is a random variable that depends on the sample S through $l_S()$. We can use the (one-sided) McDiarmid's inequality to claim that this random variable is close to its mean as n increases.

Lemma 5.8.2. *McDiarmid's inequality* [McDiarmid, 1989]: Let $z^1,...,z^n$ be n i.i.d. random variables in a set A and $h(z^1,...,z^n)$ be a function such that for all $i=1,...,n$

$$\sup_{(z^1,...,z^n,\tilde{z})\in A^{n+1}} |h(z^1,...,z^i,...,z^n) - h(z^1,...,\tilde{z},...,z^n)| \leq c.$$

$$\text{Then for all } \epsilon > 0,\ \mathbb{P}_{z^1,...,z^n}\left(h(z^1,...,z^n) - \mathbb{E}[h(z^1,...,z^n)] > \epsilon\right) \leq \exp\left(-\frac{2\epsilon^2}{nc^2}\right).$$

In our case, the function h is $\max_{f\in\mathcal{F}}|l_{\mathbb{P}}(f) - l_S(f)|$. We can show that if the i^{th} instance in the sample S is perturbed, the maximum change in the function value is $\frac{M}{n}$: We first consider the case when $\max_{f\in\mathcal{F}}|l_{\mathbb{P}}(f) - l_S(f)| \geq \max_{f\in\mathcal{F}}|l_{\mathbb{P}}(f) - l_{S^i}(f)|$.

Here $l_{S^i}(f)$ is the same as $l_S(f)$ except for the i^{th} example, which is changed from $(\mathbf{x}^i, y^i)$ to a new example $\mathbf{x}_o^i, y_o^i$. Also let $f^o \in \arg\max_{f\in\mathcal{F}} |l_{\mathbf{P}}(f) - l_S(f)|$. Then,

$$\begin{aligned}
&\max_{f\in\mathcal{F}} |l_{\mathbf{P}}(f) - l_S(f)| - \max_{f\in\mathcal{F}} |l_{\mathbf{P}}(f) - l_{S^i}(f)| \\
&\quad \le |l_{\mathbf{P}}(f^o) - l_S(f^o)| - |l_{\mathbf{P}}(f^o) - l_{S^i}(f^o)| \quad \text{(because } f^o \text{ may not maximize the second term)} \\
&\quad \le |-l_S(f^o) + l_{S^i}(f^o)| \quad \text{(by triangle inequality)} \\
&\quad = \frac{1}{n}\left| l(f^o(\mathbf{x}_o^i), y^i) - l(f^o(\mathbf{x}^i), y^i) \right| \le \frac{M}{n} \quad \text{(canceling all except the } i^{\text{th}} \text{ term).}
\end{aligned}$$

We can do an identical calculation to get the same upper bound $\frac{M}{n}$ if $\max_{f\in\mathcal{F}} |l_{\mathbf{P}}(f) - l_S(f)| \le \max_{f\in\mathcal{F}} |l_{\mathbf{P}}(f) - l_{S^i}(f)|$. Thus, with probability at least $1-\delta$,

$$\max_{f\in\mathcal{F}} |l_{\mathbf{P}}(f) - l_S(f)| \le \mathbf{E}[\max_{f\in\mathcal{F}} |l_{\mathbf{P}}(f) - l_S(f)|] + M\sqrt{\frac{\log\frac{1}{\delta}}{2n}}. \tag{5.16}$$

The quantity $\mathbf{E}[\max_{f\in\mathcal{F}} |l_{\mathbf{P}}(f) - l_S(f)|]$ captures the complexity or size of $\mathcal{F}$ (actually, its composition with the loss function l, the set $l \circ \mathcal{F}$). We can upper bound this quantity in terms of a Rademacher average using a symmetrization trick.

Lemma 5.8.3. (Upper bound)

$$\mathbf{E}[\max_{f\in\mathcal{F}} |l_{\mathbf{P}}(f) - l_S(f)|] \le 2\mathcal{R}(l \circ \mathcal{F}). \tag{5.17}$$

Proof. See Theorem 8 in Bartlett and Mendelson [2002] for essentially a similar claim.

Substituting for $\mathbf{E}[\max_{\beta\in\mathcal{B}_0} |l_{\mathbf{P}}(\beta) - l_S(\beta)|]$ from (5.17) into (5.16) gives us the desired result. □

5.8.1 Proof of Theorem 5.4.1

According to Lemma 5.8.1, the following holds with probability at least $1-\delta$ over sample S,

$$\max_{f\in\mathcal{F}} |l_{\mathbb{P}}(f) - l_S(f)| \leq 2\mathcal{R}(l\circ\mathcal{F}) + M\sqrt{\frac{\log\frac{1}{\delta}}{2n}}.$$

where $\mathcal{F}$ is a set of models, l is a bounded loss function (bounded by M), $l_{\mathbb{P}}(f) = \mathbb{E}_{\mathbf{x},y}[l(f(x),y)]$, $l_S(f) = \frac{1}{n}\sum_{i=1}^{n} l(f(\mathbf{x}^i), y^i)$ and $\mathcal{R}(l\circ\mathcal{F}) = \mathbb{E}_{S,\sigma}[\sup_{f\in\mathcal{F}} \frac{1}{n}|\sum_{i=1}^{n}\sigma^i l(f(\mathbf{x}^i), y^i)|]$.

We can apply this lemma to the case when $\mathcal{F} = \mathcal{I}$ (that is, $f(x) = I(x)$) and $l(I(x),y) = \mathbf{1}[y \notin I(x)]$. The range of the loss function is $[0,1]$, which is a bounded set. Thus, with probability at least $1-\delta$ over sample S,

$$\max_{I\in\mathcal{I}} |l_{\mathbb{P}}(I) - l_S(I)| \leq 2\mathcal{R}(l\circ\mathcal{I}) + \sqrt{\frac{\log\frac{1}{\delta}}{2n}},$$

or equivalently,

$$\forall I \in \mathcal{I} : l_S(I) - 2\mathcal{R}(l\circ\mathcal{I}) - \sqrt{\frac{\log\frac{1}{\delta}}{2n}} \leq l_{\mathbb{P}}(I) \leq l_S(I) + 2\mathcal{R}(l\circ\mathcal{I}) + \sqrt{\frac{\log\frac{1}{\delta}}{2n}}.$$

The above is a uniform convergence statement. Since it holds for I^{Alg} as well, we can state the following: with probability at least $1-\delta$ over S,

$$\begin{aligned} &l_S(I^{\text{Alg}}) - c(\delta) \leq \mathbb{P}_{\mathbf{x},y}(y \notin I^{\text{Alg}}(\mathbf{x})) \leq l_S(I^{\text{Alg}}) + c(\delta), \text{ or equivalently,} \\ &\quad 1 - \left(l_S(I^{\text{Alg}}) - c(\delta)\right) \geq \mathbb{P}_{\mathbf{x},y}(y \in I^{\text{Alg}}(\mathbf{x})) \geq 1 - \left(l_S(I^{\text{Alg}}) + c(\delta)\right), \end{aligned} \tag{5.18}$$

where $c(\delta) = 2\mathcal{R}(l\circ\mathcal{I}) + \sqrt{\frac{\log\frac{1}{\delta}}{2n}}$ and we use the relation $l_{\mathbb{P}}(I) = \mathbb{E}_{\mathbf{x},y}[\mathbf{1}[y \notin I(\mathbf{x})]] = \mathbb{P}_{\mathbf{x},y}(y \notin I(\mathbf{x}))$.

The second inequality in Equation (5.18) gives a lower bound on the probability that an unseen label belongs to the interval specified by the function $I^{\text{Alg}}(\mathbf{x})$. We can extend this lower bound to m unseen new realizations $\{\tilde{y}^j\}_{j=1}^{m}$ as follows. With probability at least $1-\delta$ over S,

$$\mathbb{P}_{\tilde{\mathbf{x}}^j,\tilde{y}^j}(\tilde{y}^j \in I^{\text{Alg}}(\tilde{\mathbf{x}}^j)) \geq 1 - \left(l_S(I^{\text{Alg}}) + c(\delta)\right) ; \; j = 1, ..., m.$$

Then, with probability $\geq 1-\delta$ over S,

$$\mathbb{P}_{\{\tilde{\mathbf{x}}^j,\tilde{y}^j\}_{j=1}^m}([\tilde{y}^1, ..., \tilde{y}^m]^T \in \Pi_{j=1}^m I^{\text{Alg}}(\tilde{\mathbf{x}}^j)) \geq (1 - \left(l_S(I^{\text{Alg}}) + c(\delta)\right))^m,$$

where we used the fact that these m events $\{\tilde{y}^j \in I^{\text{Alg}}(\tilde{\mathbf{x}}^j)\}, j = 1, ..., m$ are mutually independent given sample S.

Note that if $[\tilde{y}^1, ..., \tilde{y}^m]^T \in \Pi_{j=1}^m I^{\text{Alg}}(\tilde{\mathbf{x}}^j)$, then the robust optimal solution $\boldsymbol{\pi}^*$ is feasible for the future label realizations $\{\tilde{y}^j\}_{j=1}^m$ because it is feasible for each of the m elements in $\mathcal{U} = \Pi_{j=1}^m I^{\text{Alg}}(\tilde{\mathbf{x}}^j)$ by definition. This gives us the desired feasibility result on $\boldsymbol{\pi}^*$. □

5.8.2 Proof of Theorem 5.5.1

As in the previous proof, according to Lemma 5.8.1 the following holds with probability at least $1-\delta$ over sample S:

$$\max_{f \in \mathcal{F}} |l_{\mathbb{P}}(f) - l_S(f)| \leq 2\mathcal{R}(l \circ \mathcal{F}) + M\sqrt{\frac{\log \frac{1}{\delta}}{2n}}.$$

where $\mathcal{F}$ is a set of models, l is a bounded loss function (bounded by M), $l_{\mathbb{P}}(f) = \mathbb{E}_{\mathbf{x},y}[l(f(x), y)]$, $l_S(f) = \frac{1}{n}\sum_{i=1}^n l(f(\mathbf{x}^i), y^i)$ and

$$\mathcal{R}(l \circ \mathcal{F}) = \mathbb{E}_{S,\sigma}\left[\sup_{f \in \mathcal{F}} \frac{1}{n}\left|\sum_{i=1}^n \sigma^i l(f(\mathbf{x}^i), y^i)\right|\right].$$

We will apply the lemma in two cases. For both cases, let the model set $\mathcal{B}_0$ be the set of conditional quantile models. For the first case, let the loss function be $r_\epsilon^-(y - \beta(\mathbf{x}))$ and for the second case, let the loss function be $r_\epsilon^+(y - \beta(\mathbf{x}))$ (both functions are defined in the statement of Theorem 5.5.1). The range of both functions is $[0, 1]$, and thus bounded. Further, since Lemma 5.8.1 is a uniform deviation statement, the inequality also holds for model $\beta^{\mathrm{Alg},\tau}$, derived from sample S (say, by minimizing the pinball loss), with probability at least $1-\delta$. Thus, we have the following two probabilistic statements:

- With prob. $\geq 1-\delta$ over S,

$$\left|\mathbb{E}_{\mathbf{x},y}[r_\epsilon^-(y-\beta^{\mathrm{Alg},\tau}(\mathbf{x}))]-\frac{1}{n}\sum_{i=1}^{n}r_\epsilon^-(y^i-\beta^{\mathrm{Alg},\tau}(\mathbf{x}^i))\right|\leq 2\mathcal{R}(r_\epsilon^-\circ\mathcal{B}_0)+\sqrt{\frac{\log\frac{1}{\delta_2}}{2n}}. \tag{5.19}$$

- With prob. $\geq 1-\delta$ over S,

$$\left|\mathbb{E}_{\mathbf{x},y}[r_\epsilon^+(y-\beta^{\mathrm{Alg},\tau}(\mathbf{x}))]-\frac{1}{n}\sum_{i=1}^{n}r_\epsilon^+(y^i-\beta^{\mathrm{Alg},\tau}(\mathbf{x}^i))\right|\leq 2\mathcal{R}(r_\epsilon^+\circ\mathcal{B}_0)+\sqrt{\frac{\log\frac{1}{\delta_2}}{2n}}. \tag{5.20}$$

From these inequalities, we get the following lemma [similar to Takeuchi et al., 2006, Theorem 7]:

Lemma 5.8.4. With probability at least $1-\delta$ over sample S, the following inequalities hold separately:

$$\frac{1}{n}\sum_{i=1}^{n}r_\epsilon^-(y^i-\beta^{\mathrm{Alg},\tau}(\mathbf{x}^i))-c\leq\mathbb{P}_{\mathbf{x},y}(y\leq\beta^{\mathrm{Alg},\tau}(\mathbf{x})),\ \text{and} \tag{5.21}$$

$$\mathbb{P}_{\mathbf{x},y}(y\leq\beta^{\mathrm{Alg},\tau}(\mathbf{x}))\leq\frac{1}{n}\sum_{i=1}^{n}r_\epsilon^+(y^i-\beta^{\mathrm{Alg},\tau}(\mathbf{x}^i))+c, \tag{5.22}$$

where $c:=\frac{4}{\epsilon}\mathcal{R}(\mathcal{B}_0)+\sqrt{\frac{\log\frac{1}{\delta}}{2n}}$.

Proof (of Lemma 5.8.4) From the Ledoux-Talagrand contraction inequality, we know $\mathcal{R}(l\circ\mathcal{B}_0)\leq 2\mathcal{L}\mathcal{R}(\mathcal{B}_0)$. In our case, both r_ϵ^- and r_ϵ^+ have Lipschitz constant equal to $1/\epsilon$. Let c be defined as in the statement of the lemma. From the inequalities

(5.19) and (5.20) we get the one sided inequalities:

$$\mathbb{E}_{\mathbf{x},y}[r_\epsilon^-(y-\beta^{\mathrm{Alg},\tau}(\mathbf{x}))] \geq \frac{1}{n}\sum_{i=1}^{n} r_\epsilon^-(y^i-\beta^{\mathrm{Alg},\tau}(\mathbf{x}^i)) - c, \text{ and}$$

$$\mathbb{E}_{\mathbf{x},y}[r_\epsilon^+(y-\beta^{\mathrm{Alg},\tau}(\mathbf{x}))] \leq \frac{1}{n}\sum_{i=1}^{n} r_\epsilon^+(y^i-\beta^{\mathrm{Alg},\tau}(\mathbf{x}^i)) + c.$$

Further, for any β, we can bound $\mathbb{P}_{\mathbf{x},y}(y \leq \beta(\mathbf{x})) = \mathbb{E}_{\mathbf{x},y}[1[y \leq \beta(\mathbf{x})]]$ from both sides because of the following inequalities:

$$\mathbb{E}_{\mathbf{x},y}[1[y \leq \beta(\mathbf{x})]] \leq \mathbb{E}_{\mathbf{x},y}[r_\epsilon^+(y-\beta(\mathbf{x}))], \text{ and} \tag{5.23}$$

$$\mathbb{E}_{\mathbf{x},y}[1[y \leq \beta(\mathbf{x})]] \geq \mathbb{E}_{\mathbf{x},y}[r_\epsilon^-(y-\beta(\mathbf{x}))]. \tag{5.24}$$

Thus we get:

- with prob. $\geq 1-\delta$, $\mathbb{P}_{\mathbf{x},y}(y \leq \beta^{\mathrm{Alg},\tau}(\mathbf{x})) \geq \frac{1}{n}\sum_{i=1}^{n} r_\epsilon^-(y^i-\beta^{\mathrm{Alg},\tau}(\mathbf{x}^i)) - c$, and

- with prob. $\geq 1-\delta$, $\mathbb{P}_{\mathbf{x},y}(y \leq \beta^{\mathrm{Alg},\tau}(\mathbf{x})) \leq \frac{1}{n}\sum_{i=1}^{n} r_\epsilon^+(y^i-\beta^{\mathrm{Alg},\tau}(\mathbf{x}^i)) + c$.

□

Continuing with the proof of Theorem 5.5.1, we apply Lemma 5.8.4 with $\tau = \delta_p$ within inequality (5.22) and with $\tau = \delta_q$ within inequality (5.21), where $1 \leq \delta_p < \delta_q \leq 1$ to obtain:

- with prob. $\geq 1-\delta$, $\mathbb{P}_{\mathbf{x},y}(y \leq \beta^{\mathrm{Alg},\delta_q}(\mathbf{x})) \geq \frac{1}{n}\sum_{i=1}^{n} r_\epsilon^-(y^i-\beta^{\mathrm{Alg},\delta_q}(\mathbf{x}^i)) - c$, and

- with prob. $\geq 1-\delta$, $\mathbb{P}_{\mathbf{x},y}(y \leq \beta^{\mathrm{Alg},\delta_p}(\mathbf{x})) \leq \frac{1}{n}\sum_{i=1}^{n} r_\epsilon^+(y^i-\beta^{\mathrm{Alg},\delta_p}(\mathbf{x}^i)) + c$.

The bounds hold with probabilities $1-\delta$ each implying that they together hold with probability $1 - 2\delta$. Now,

$$
\begin{aligned}
&\mathbb{P}_{\mathbf{x},y}(\beta^{\text{Alg},\delta_p}(\mathbf{x}) < y \leq \beta^{\text{Alg},\delta_q}(\mathbf{x})) \\
&\quad = \mathbb{P}_{\mathbf{x},y}(\{\beta^{\text{Alg},\delta_p}(\mathbf{x}) < y\} \cap \{y \leq \beta^{\text{Alg},\delta_q}(\mathbf{x})\}) \\
&\quad = 1 - \mathbb{P}_{\mathbf{x},y}(\{y \leq \beta^{\text{Alg},\delta_p}(\mathbf{x})\} \cup \{\beta^{\text{Alg},\delta_q}(\mathbf{x}) < y\}) \\
&\quad \geq 1 - \big(\mathbb{P}_{\mathbf{x},y}(y \leq \beta^{\text{Alg},\delta_p}(\mathbf{x})) + \mathbb{P}_{\mathbf{x},y}(\beta^{\text{Alg},\delta_q}(\mathbf{x}) < y)\big) \\
&\quad = 1 - \big(\mathbb{P}_{\mathbf{x},y}(y \leq \beta^{\text{Alg},\delta_p}(\mathbf{x})) + 1 - \mathbb{P}_{\mathbf{x},y}(y \leq \beta^{\text{Alg},\delta_q}(\mathbf{x}))\big) \\
&\quad = \mathbb{P}_{\mathbf{x},y}(y \leq \beta^{\text{Alg},\delta_q}(\mathbf{x})) - \mathbb{P}_{\mathbf{x},y}(y \leq \beta^{\text{Alg},\delta_p}(\mathbf{x})) \\
&\quad \overset{(*)}{\geq} \frac{1}{n}\sum_{i=1}^{n} r_\epsilon^-(y^i - \beta^{\text{Alg},\delta_q}(\mathbf{x}^i)) - \frac{1}{n}\sum_{i=1}^{n} r_\epsilon^+(y^i - \beta^{\text{Alg},\delta_p}(\mathbf{x}^i)) - 2c,
\end{aligned}
$$

where in step $(*)$, we substituted upper and lower bounds of the two random variables of S, $\mathbb{P}_{\mathbf{x},y}(y \leq \beta^{\text{Alg},\delta_q}(\mathbf{x}))$ and $\mathbb{P}_{\mathbf{x},y}(y \leq \beta^{\text{Alg},\delta_p}(\mathbf{x}))$. Thus, with probability $\geq 1 - 2\delta$ over S,

$$
\mathbb{P}_{\mathbf{x},y}(y \in [\beta^{\text{Alg},\delta_p}(\mathbf{x}), \beta^{\text{Alg},\delta_q}(\mathbf{x})]) \geq \frac{1}{n}\sum_{i=1}^{n} \big(r_\epsilon^-(y^i - \beta^{\text{Alg},\delta_q}(\mathbf{x}^i)) - r_\epsilon^+(y^i - \beta^{\text{Alg},\delta_p}(\mathbf{x}^i))\big) - 2c.
$$

In the above statement, we have a lower bound on the probability that a new unseen realization y belongs to the random interval $[\beta^{\text{Alg},\delta_p}(\mathbf{x}), \beta^{\text{Alg},\delta_q}(\mathbf{x})]$.

We can extend this lower bound to the setting of m simultaneous lower bounds corresponding to m unseen new realizations $\{\tilde{y}^j\}_{j=1}^m$ in our decision problem as follows. We know that with probability $\geq 1 - 2\delta$ over S,

$$
\mathbb{P}_{\tilde{\mathbf{x}}^j,\tilde{y}^j}(\tilde{y}^j \in [\beta^{\text{Alg},\delta_p}(\tilde{\mathbf{x}}^j), \beta^{\text{Alg},\delta_q}(\tilde{\mathbf{x}}^j)]) \geq \Delta(S);\ j = 1, ..., m,
$$

where $\Delta(S) := \frac{1}{n}\sum_{i=1}^{n} \big(r_\epsilon^-(y^i - \beta^{\text{Alg},\delta_q}(\mathbf{x}^i)) - r_\epsilon^+(y^i - \beta^{\text{Alg},\delta_p}(\mathbf{x}^i))\big) - 2c$. Then, with probability $\geq 1 - 2\delta$ with respect to sample S,

$$
\mathbb{P}_{\{\tilde{\mathbf{x}}^j,\tilde{y}^j\}_{j=1}^m}([\tilde{y}^1, ..., \tilde{y}^m]^T \in \Pi_{j=1}^m [\beta^{\text{Alg},\delta_p}(\tilde{\mathbf{x}}^j), \beta^{\text{Alg},\delta_q}(\tilde{\mathbf{x}}^j)]) \geq \Delta(S)^m,
$$

where we used the fact that these m events $\{\tilde{y}^j \in [\beta^{\text{Alg},\delta_p}(\tilde{\mathbf{x}}^j), \beta^{\text{Alg},\delta_q}(\tilde{\mathbf{x}}^j)]\}, j = 1, ..., m$

are mutually independent given sample S.

Note that if $[\tilde{y}^1, ..., \tilde{y}^m]^T \in \Pi_{j=1}^m [\beta^{\text{Alg},\delta_p}(\tilde{\mathbf{x}}^j), \beta^{\text{Alg},\delta_q}(\tilde{\mathbf{x}}^j)]$, then it also belongs to $\mathcal{U}$ defined by Equation (5.5). Further, the robust optimal solution π^* will be feasible for $\{\tilde{y}^j\}_{j=1}^m$ because it is feasible for every element in $\mathcal{U}$ by definition. Thus, changing δ to $\delta/2$ (with an appropriate change in the constant c in Equations (5.21) and (5.22)) gives us the desired feasibility result on π^*. □

5.8.3 Proof of Theorem 5.6.1

Consider the term $l_S(\beta^*) - l_S(\beta^{Alg})$, which depends on the random sample S. We can upper bound it by:

$$\begin{aligned}
l_S(\beta^*)-&l_S(\beta^{Alg}) \\
&= l_S(\beta^*) - l_{\mathbf{P}}(\beta^*) + l_{\mathbf{P}}(\beta^*) - l_S(\beta^{Alg}) \\
&\leq l_S(\beta^*) - l_{\mathbf{P}}(\beta^*) + l_{\mathbf{P}}(\beta^{Alg}) - l_S(\beta^{Alg}) \\
&\leq l_S(\beta^*) - l_{\mathbf{P}}(\beta^*) + \max_{\beta \in \mathcal{B}_0} |l_{\mathbf{P}}(\beta) - l_S(\beta)| \qquad (5.25)
\end{aligned}$$

where we added and subtracted $l_{\mathbf{P}}(\beta^*)$ in the first step, then in the second step substituted β^{Alg} for β^* in the third term to increase the value of the right hand side, and finally in the last step, replaced the last two terms with an absolute max operation over $\mathcal{B}_0$.

The first term in the expression on the right hand side of (5.25) will go to zero in probability as $n \to \infty$ due to concentration, and this can be quantified for finite n via Hoeffding's inequality.

Lemma 5.8.5. *(One-sided Hoeffding's inequality.)* Let $z^1, ..., z^n$ and z be i.i.d. random variables and let h be a bounded function, $a \leq h(z) \leq b$. Then for all $\epsilon > 0$ we have

$$\mathbb{P}_{z^1,...,z^n}\left(\frac{1}{n}\sum_{i=1}^n h(z^i) - \mathbb{E}_z[h(z)] > \epsilon\right) \leq \exp\left(-\frac{2n\epsilon^2}{(b-a)^2}\right).$$

In our case, the sample S is represented by $\{(\mathbf{x}^i, y^i)\}_{i=1}^n$. The function $l(\beta^*(x), y)$ is bounded in the interval $[0, M]$. Thus the empirical mean $\frac{1}{n}\sum_{i=1}^n l(\beta^*(x^i), y^i)$ ($= l_S(\beta^*)$) gets close to its mean $\mathbb{E}[l(\beta^*(x), y)]$ ($= l_{\mathbf{P}}(\beta^*)$) as n increases. In particular, we see that with probability at least $1 - \delta_1$,

$$l_S(\beta^*) - l_{\mathbf{P}}(\beta^*) \leq M\sqrt{\frac{\log \frac{1}{\delta_1}}{2n}}. \tag{5.26}$$

The second term (5.25) can be bounded using Lemma 5.8.1 which states that with probability at least $1 - \delta$ over sample S,

$$\max_{f \in \mathcal{F}} |l_{\mathbf{P}}(f) - l_S(f)| \leq 2\mathcal{R}(l \circ \mathcal{F}) + M\sqrt{\frac{\log \frac{1}{\delta}}{2n}}.$$

In our case, we set $\mathcal{F} = \mathcal{B}_0$ and $f(x) = \beta(x)$ and $\delta = \delta_2$.

The empirical Rademacher average $\mathcal{R}_S(l \circ \mathcal{B}_0)$ also concentrates around its mean $\mathcal{R}(l \circ \mathcal{B}_0)$ and this can be proved again by McDiarmid's inequality. In this case, from Lemma 5.8.2, the function h is represented by $\mathcal{R}_S(l \circ \mathcal{B}_0)$. We can again show [Bartlett and Mendelson, 2002, Theorem 11] that if the i^{th} instance in the sample S is perturbed, the maximum change in the function value is $\frac{M}{n}$. Thus, with probability at least $1 - \delta_3$,

$$\mathcal{R}(l \circ \mathcal{B}_0) \leq \mathcal{R}_S(l \circ \mathcal{B}_0) + M\sqrt{\frac{\log \frac{1}{\delta_3}}{2n}}. \tag{5.27}$$

In summary we have the following statements for the terms on the right hand side of (5.25):

1. With probability at least $1 - \delta_1$ over S, $l_S(\beta^*) - l_{\mathbf{P}}(\beta^*) \leq M\sqrt{\frac{\log \frac{1}{\delta_1}}{2n}}$ from (5.26).

2. With probability at least $1 - \delta_2$ over S,

$$\max_{\beta \in \mathcal{B}_0} |l_{\mathbf{P}}(\beta) - l_S(\beta)| \leq 2\mathcal{R}(l \circ \mathcal{B}_0) + M\sqrt{\frac{\log \frac{1}{\delta_2}}{2n}}.$$

3. With probability at least $1-\delta_3$ over S, $\mathcal{R}(l \circ \mathcal{B}_0) \leq \mathcal{R}_S(l \circ \mathcal{B}_0) + M\sqrt{\frac{\log \frac{1}{\delta_3}}{2n}}$ from (5.27).

Consider the three corresponding events: $E_1 = \left\{S \ : \ l_S(\beta^*) - l_{\mathbb{P}}(\beta^*) \leq M\sqrt{\frac{\log \frac{1}{\delta_1}}{2n}}\right\}$, $E_2 = \left\{S \ : \ \max_{\beta \in \mathcal{B}_0} (l_{\mathbb{P}}(\beta) - l_S(\beta)) \leq 2\mathcal{R}(l \circ \mathcal{B}_0) + M\sqrt{\frac{\log \frac{1}{\delta_2}}{2n}}\right\}$, and $E_3 = \left\{S \ : \ \mathcal{R}(l \circ \mathcal{B}_0) \leq \mathcal{R}_S(l \circ \mathcal{B}_0) + M\sqrt{\frac{\log \frac{1}{\delta_3}}{2n}}\right\}$. We know that with probabilities $\delta_1, \delta_2, \delta_3$ over the random sample S, these events do not happen. Thus using the union bound, $\mathbb{P}_S(E_1 \cap E_2 \cap E_3) \geq 1 - \delta_1 + \delta_2 + \delta_3$. Substituting $\frac{\delta}{3}$ for δ_1, δ_2 and δ_3 and using (5.25), we that with probability at least $1-\delta$,

$$l_S(\beta^*) - l_S(\beta^{Alg}) \leq 2\mathcal{R}_S(l \circ \mathcal{B}_0) + 4M\sqrt{\frac{\log \frac{3}{\delta}}{2n}}.$$

The implication of this is that the empirical risk for the 'best-in-class' function β^* is less than the right hand side quantities, all of which are computable. This implies that even though we do not know β^*, we know it belongs to our uncertainty set precursor $\mathcal{B}$ defined in Equation (5.11) with high probability. In particular, we see that $\beta^* \in \mathcal{B}$ with probability at least $1-\delta$ over sample S. This is part (1) in the statement of the Theorem.

Part (1) further implies that with probability at least $1-\delta$, $u_{\beta^*} \in \mathcal{U}_{\mathcal{B}}$, and this is true for any $\{\tilde{\mathbf{x}}^j\}_{j=1}^m$. Next we turn our focus toward model residuals. We can extend the probabilistic statement in Equation (5.6) to the setting where we have m simultaneous errors using the mutual independence assumption. Thus we have, with probability at least $(1-\delta_e)^m$ over $\{(\tilde{\mathbf{x}}^j, \tilde{y}^j)\}_{j=1}^m$, $\max_{j=1,\ldots,m} |\tilde{y}^j - \beta^*(\tilde{\mathbf{x}}^j)| \in E$. Using the definition of set $\mathcal{U}_{-\mathcal{B}}$, which is equal to E^m, we see that $u_{-\beta*} \in \mathcal{U}_{-\mathcal{B}}$ with probability at least $(1-\delta_e)^m$ over $\{(\tilde{\mathbf{x}}^j, \tilde{y}^j)\}_{j=1}^m$.

We know that the robust optimal solution $\boldsymbol{\pi}^*$ of Equation (5.2) is robust to any element of $\mathcal{U} = \mathcal{U}_{\mathcal{B}} + \mathcal{U}_{-\mathcal{B}}$ by definition. In particular, if $\beta^* \in \mathcal{B}$ and $u_{-\beta^*} \in \mathcal{U}_{-\mathcal{B}}$, then $\boldsymbol{\pi}^*$ will be robust to the random vector $\mathbf{u}_{\beta^*} + \mathbf{u}_{-\beta^*}$ (which equals $[\tilde{y}^1 \ldots \tilde{y}^m]^T$).

To get a guarantee of robustness of π^* to $\{\tilde{y}^j\}_{j=1}^m$, we can combine the two probabilistic statements above (one with respect to S and the other with respect to $\{(\tilde{\mathbf{x}}^j, \tilde{y}^j)\}_{j=1}^m$) using the mutual independence assumption (S and $\{(\tilde{\mathbf{x}}^j, \tilde{y}^j)\}_{j=1}^m$ are mutually independent) as follows:

$$\mathbb{P}_{S,\{(\tilde{\mathbf{x}}^j,\tilde{y}^j)\}_{j=1}^m}\left(F(\pi^*, [\tilde{y}^1 \dots \tilde{y}^m]^T) \in \mathcal{K}\right) \geq (1-\delta)(1-\delta_e)^m. \quad \Box$$

5.8.4 Proof of Theorem 5.6.2

It is sufficient to show that with probability at least $1-\delta$, $\beta^* \in \mathcal{B}$ where $\mathcal{B}$ is defined in Equation (5.12). To see this, consider the deviation $l_S(\beta^*) - l_S(\beta^{Alg})$. This can be upper bounded in a similar way as in the beginning of the proof of Theorem 5.6.1:

$$l_S(\beta^*) - l_S(\beta^{Alg}) \leq l_S(\beta^*) - l_{\mathbf{P}}(\beta^*) + \max_{\beta\in\mathcal{B}_0}(l_{\mathbf{P}}(\beta) - l_S(\beta)).$$

We will upper bound the two deviation terms appearing on the right hand side of the above inequality. Both terms are functions of the random sample S.

Lets begin with the term $\max_{\beta\in\mathcal{B}_0}(l_{\mathbf{P}}(\beta) - l_S(\beta))$. We can bound the probability of the event $\{\max_{\beta\in\mathcal{B}_0}(l_{\mathbf{P}}(\beta) - l_S(\beta)) > \epsilon\}$ as follows:

$$\begin{aligned}\mathbb{P}_S\Big(\max_{\beta\in\mathcal{B}_0}(l_{\mathbf{P}}(\beta) - l_S(\beta)) > \epsilon\Big) &= \mathbb{P}_S\Big(\cup_{i=1}^{|\mathcal{B}_0|}\{l_{\mathbf{P}}(\beta^i) - l_S(\beta^i) > \epsilon\}\Big)\\ &\overset{(a)}{\leq} \sum_{i=1}^{|\mathcal{B}_0|}\mathbb{P}_S\Big(l_{\mathbf{P}}(\beta^i) - l_S(\beta^i) > \epsilon\Big) \overset{(b)}{=} \sum_{i=1}^{|\mathcal{B}_0|} e^{-\frac{2n\epsilon^2}{M^2}} = e^{\log|\mathcal{B}_0| - \frac{2n\epsilon^2}{M^2}}.\end{aligned}$$

Here, (a) follows from taking a union bound, and (b) follows from applying Hoeffding's inequality to each fixed model $\beta^i, i = 1, ..., |\mathcal{B}_0|$. Setting $\delta_2 = e^{\log|\mathcal{B}_0| - \frac{2n\epsilon^2}{M^2}}$ and replacing ϵ gives us the following equivalent way to state the same result: with probability at least $1-\delta_2$ over S,

$$\max_{\beta\in\mathcal{B}_0}(l_{\mathbf{P}}(\beta) - l_S(\beta)) \leq M\sqrt{\frac{\log|\mathcal{B}_0| + \log(\frac{1}{\delta_2})}{2n}}.$$

From Equation (5.26), we have the following upper bound for the term $l_S(\beta^*) -$

$l_{\mathbb{P}}(\beta^*)$: with probability at least $1 - \delta_1$ over S, $l_S(\beta^*) - l_{\mathbb{P}}(\beta^*) \leq M\sqrt{\frac{\log \frac{1}{\delta_1}}{2n}}$.

Using a union bound with these two observations gives us the following statement when we set $\delta_1 = \delta_2 = \delta/2$: with probability at least $1-\delta$ over S, $l_S(\beta^*) - l_S(\beta^{Alg}) \leq M\sqrt{\frac{\log|\mathcal{B}_0| + \log(\frac{2}{\delta})}{2n}} + M\sqrt{\frac{\log \frac{2}{\delta}}{2n}}$. Thus $\beta^* \in \mathcal{B}$ with probability at least $1 - \delta$ as desired.

□

5.8.5 Proof of Theorem 5.7.1

Proof of part (1) is the same as that of part (1) in Theorem 5.6.1. That is, using the definition of $\mathcal{B}^\tau$ in Equation (5.15) and Lemma 5.8.1 with the pinball loss function l^τ we see that $\beta^{\tau,*} \in \mathcal{B}^\tau$ with probability at least $1 - \delta$ over S. Thus, part (1) holds when τ is set to δ_p and δ_q individually.

For part (2), we use mutual independence and union bound arguments, similar to part (2) in Theorem 5.6.1. In particular,

- With prob. $\geq 1 - \delta$ over S, simultaneously for all $j = 1,..,m$, $\beta^{\delta_p,*}(\tilde{\mathbf{x}}^j) \in [\inf\{\beta(\tilde{\mathbf{x}}^j) : \beta \in \mathcal{B}^{\delta_p}\}, \sup\{\beta(\tilde{\mathbf{x}}^j) : \beta \in \mathcal{B}^{\delta_p}\}]$ for any $\{\tilde{\mathbf{x}}^j\}_{j=1}^m$ (from part (1)).

- With prob. $\geq 1 - \delta$ over S, simultaneously for all $j = 1,..,m$, $\beta^{\delta_q,*}(\tilde{\mathbf{x}}^j) \in [\inf\{\beta(\tilde{\mathbf{x}}^j) : \beta \in \mathcal{B}^{\delta_q}\}, \sup\{\beta(\tilde{\mathbf{x}}^j) : \beta \in \mathcal{B}^{\delta_q}\}]$ for any $\{\tilde{\mathbf{x}}^j\}_{j=1}^m$ (from part (1)).

- With prob. $\geq (1 - \delta_e^{\delta_p})^m$ over $\{(\tilde{\mathbf{x}}^j, \tilde{y}^j)\}_{j=1}^m$, simultaneously for all $j = 1,..,m$, $\mu^{\delta_p}(\tilde{\mathbf{x}}^j) - \beta^{\delta_p,*}(\tilde{\mathbf{x}}^j) \in [-\sup E^{\delta_p}, \sup E^{\delta_p}]$ (using mutual independence assumption and Equation (5.8)).

- With prob. $\geq (1 - \delta_e^{\delta_q})^m$ over $\{(\tilde{\mathbf{x}}^j, \tilde{y}^j)\}_{j=1}^m$, simultaneously for all $j = 1,..,m$, $\mu^{\delta_q}(\tilde{\mathbf{x}}^j) - \beta^{\delta_q,*}(\tilde{\mathbf{x}}^j) \in [-\sup E^{\delta_q}, \sup E^{\delta_q}]$ (using mutual independence assumption and Equation (5.8)).

We can again use the mutual independence between S and $\{(\tilde{\mathbf{x}}^j, \tilde{y}^j)\}_{j=1}^m$ to claim the following:

- With prob. $\geq (1 - \delta)(1 - \delta_e^{\delta_p})^m$ over S and $\{(\tilde{\mathbf{x}}^j, \tilde{y}^j)\}_{j=1}^m$, simultaneously for all $j = 1, ..., m$, $\mu^{\delta_p}(\tilde{\mathbf{x}}^j) \in [\inf\{\beta(\tilde{\mathbf{x}}^j) : \beta \in \mathcal{B}^{\delta_p}\} - \sup E^{\delta_p}, \sup\{\beta(\tilde{\mathbf{x}}^j) : \beta \in \mathcal{B}^{\delta_p}\} + \sup E^{\delta_p}]$.

- With prob. $\geq (1-\delta)(1-\delta_e^{\delta_q})^m$ over S and $\{(\tilde{\mathbf{x}}^j, \tilde{y}^j)\}_{j=1}^m$, simultaneously for all $j = 1, ..., m$, $\mu^{\delta_q}(\tilde{\mathbf{x}}^j) \in [\inf\{\beta(\tilde{\mathbf{x}}^j) : \beta \in \mathcal{B}^{\delta_q}\} - \sup E^{\delta_q}, \sup\{\beta(\tilde{\mathbf{x}}^j) : \beta \in \mathcal{B}^{\delta_q}\} + \sup E^{\delta_q}]$.

We use the general identity from De Morgan's laws and the union bound that if $\mathbb{P}(A_1) \geq c_1$ and $\mathbb{P}(A_2) \geq c_2$, then $\mathbb{P}(A_1 \cap A_2) \geq c_1 + c_2 - 1$. Applying this to the two events above, we see that with probability at least $(1-\delta)\left[(1-\delta_e^{\delta_p})^m + (1-\delta_e^{\delta_q})^m\right] - 1$ over S and $\{(\tilde{\mathbf{x}}^j, \tilde{y}^j)\}_{j=1}^m$,

$$
\begin{aligned}
&[\mu^{\delta_p}(\tilde{\mathbf{x}}^j), \mu^{\delta_q}(\tilde{\mathbf{x}}^j)] \subseteq \\
&\qquad [\inf\{\beta(\tilde{\mathbf{x}}^j) : \beta \in \mathcal{B}^{\delta_p} \cup \mathcal{B}^{\delta_q}\} - \sup E^{\delta_p} \cup E^{\delta_q}, \\
&\qquad \sup\{\beta(\tilde{\mathbf{x}}^j) : \beta \in \mathcal{B}^{\delta_p} \cup \mathcal{B}^{\delta_q}\} + \sup E^{\delta_p} \cup E^{\delta_q}].
\end{aligned}
$$

We also know that simultaneously for all j, $\tilde{y}^j$ belongs to $[\mu^{\delta_p}(\tilde{\mathbf{x}}^j), \mu^{\delta_q}(\tilde{\mathbf{x}}^j)]$ with probability at least $(\delta_q - \delta_p)^m$ over $\{(\tilde{\mathbf{x}}^j, \tilde{y}^j)\}_{j=1}^m$ (mutual independence and definition of conditional quantile function). Thus, again using the identity based on De Morgan's laws and the union bound, we get that with probability at least $(1-\delta)\left[(1-\delta_e^{\delta_p})^m + (1-\delta_e^{\delta_q})^m\right] + (\delta_q - \delta_p)^m - 2$ over S and $\{(\tilde{\mathbf{x}}^j, \tilde{y}^j)\}_{j=1}^m$, $[\tilde{y}^1 ... \tilde{y}^m]^T$ belongs to the set

$$
\begin{aligned}
\Pi_{j=1}^m \Big[&\inf\{\beta(\tilde{\mathbf{x}}^j) : \beta \in \mathcal{B}^{\delta_p} \cup \mathcal{B}^{\delta_q}\} - \sup E^{\delta_p} \cup E^{\delta_q}, \\
&\sup\{\beta(\tilde{\mathbf{x}}^j) : \beta \in \mathcal{B}^{\delta_p} \cup \mathcal{B}^{\delta_q}\} + \sup E^{\delta_p} \cup E^{\delta_q}\Big].
\end{aligned}
$$

Since $\mathcal{U}$ is defined precisely using the above product set, we conclude that the robust optimal solution $\boldsymbol{\pi}^*$ is feasible for $\{\tilde{y}^j\}_{j=1}^m$ with the desired guarantee. □

5.9 Conclusion

In this work, we presented two principled approaches (four methods) of constructing uncertainty sets for robust optimization based on statistical learning theory. These methods can be used broadly for data-driven robust optimization, and apply to any problem where the data are drawn from an unknown distribution. The first two methods can be applied without any distributional assumptions, and the other two methods require very mild distributional assumptions, which is that the user knows one statistic about the tail of the distribution. The results in this chapter show that statistical learning theory, derived for guarantees on prediction quality of statistical models, can be used for guarantees on the robustness of an optimization problem.

Chapter 6

Tire Changes, Fresh Air, and Yellow Flags: Challenges in Predictive Analytics for Professional Racing

6.1 Introduction

Currently in the United States, professional car racing has the second largest viewing audience among all sports[1]. Within a professional stock car race, some of the most critical decisions by the teams are made during pit-stops, where teams can choose to change either zero tires, two tires or all four tires of their car. Changing four tires is more time consuming, and teams can risk losing their advantage over the other players because of extra time spent changing tires in the pit; on the other hand, changing two tires or zero tires may be risky, since providing the car with fewer fresh tires could decrease its maximum potential speed. Predicting in advance which decision would most benefit a team can depend on many complex variables, a relationship that is difficult for racing teams to predict. Currently the choice needs to be made by the team captain instantaneously, without computational tools, yet somehow considering all possible data about each team in the race. These are key decisions, viewed by

[1] http://www.shavemagazine.com/cars/090601 Shave Magazine "'All About NASCAR" by Kiley Alderink

millions of fans, that are made almost purely from experience and judgment rather than with the help of analytical tools.

There are many other sports in which key strategic decisions are made without the help of in-game analytical tools. Even in sports like baseball and basketball, where there has been a lot of work on analytics, analyses are typically done at the season level, prior to the start of the game. This is very different than our work. This is because, in racing, the actual conditions of the race are potentially very useful for predicting the outcomes, beyond what one can obtain using season level statistics.

This work started with the hypothesis that a data-driven prediction engine operating in real-time may be able to assist team captains in making these critical tire-change decisions. As no such prediction software or methodology previously existed to do this, it was unclear how the data could be leveraged to produce an accurate prediction model; there was no previous knowledge discovery system for working with data from professional stock car races, or from any similar enough sport. Further, the predictions need to be made at the finest granularity available for racing data - at the level of individual laps - which is the most detailed race-level data made available to teams by NASCAR (at least through 2012). While constructing a knowledge discovery system for these data, we faced considerable challenges in how to process and define the prediction model. In handling racing data, it is easy for a bad mathematical definition to lead to a conclusion that a particular feature is not important for prediction, and it is easy for Simpson's paradox to appear, indicating (for instance) that tire change decisions do not impact race position. In the end, we were able to obtain high quality results only when domain expert knowledge about racing was carefully infused into all of the mathematically defined features and evaluation metrics used in the prediction engine.

We consider the entire cycle of the knowledge discovery process: exploratory analysis, feature generation, building a model, data mining, and decision making for within-race strategy. Mining the raw data requires many domain specific considerations in order to construct meaningful statistics. Model building requires careful assumptions about the observed data, and molding the problem into a tractable learning formulation. Based on the model outputs, decision making requires an understanding of the horizon and time scale where is it most meaningful to make a decision and characterize its risk-reward tradeoff. In the sports prediction and decision making studies done in the past, these components have been examined mainly in isolation. Our study can be abstracted to a framework that is both unified and tractable, allowing the possibility of system-optimal solutions in a practical amount of time (instantaneously) for professional racing and other sports.

The statistical hypotheses we address will be derived from the following questions:

Q1. Can we predict the change in rank position of a racer over the next portion of the race, based on the racers' recent history?

Q2. Can we optimize within-race tire change and refueling strategy, based on the predicted future performance of a racer?

Q3. Can we gain insight from past races that can assist the team for a future race?

Considering question **Q1**, the design of in-race data-driven strategy critically relies on our ability to forecast the performance of the racer based on his and his neighbors' recent race history, the state of the race up to that point, and any decisions he can potentially make (zero tires, two tires and four tires). The racer's recent history can include the number of other racers he overtook, the racer's speed, rank position, and the age of each of his tires. Another valuable outcome of answering **Q1** is to be able to forecast the finishing rank as early as possible within the race. This is conventionally forecasted using season level data, before the race even starts.

To determine strategy, we need to know beforehand what the impact of a racer's tire change will be on his rank position and deceleration. It is possible for a racer to rapidly gain rank position by changing zero or two tires during a pit stop, but this action can penalize his ability to maintain this rank position throughout the next portion of the race. This effect can be highly complex, and dependent not just on the racer, but on the tire-change decisions of other racers, the track itself, the track temperature and weather, and the type of tire used for the race. Yet, being able to forecast the impact of a tire change decision can assist with critical elements of racing strategy; in other words, answering **Q1** can lead to an answer for **Q2**. For instance, a reasonable myopic strategy is as follows: if we predict that a two tire change is likely to lead to a loss in track position compared to a four tire change, the team captain could make a decision to change four tires. Answering **Q2** is important since strategies may have a large impact on the racer's success when all his peers are almost equally skilled and the cars have very comparable speeds.

Besides the goals of real-time prediction and decision making, a knowledge discovery framework for racing can help to provide specific insights into racing strategy (**Q3**). It can be a valuable tool for reasoning about how different actions in the past have impacted the subsequent rank positions of the racers. For instance, does the value of the prediction depend on the forecast horizon? Does the variability of laps raced between tire changes have an effect on ranks? We would like to know answers to such questions because they can lead to better predictions and insight for future races.

Section 6.2 provides related work. In Section 6.3, we describe some of the complexities we encountered in the knowledge discovery process in our setting. We also describe some experimental shortcomings that restrict the predictions and inferences we can make. In Section 6.4, we define the prediction problem and describe the key hypotheses about our data that guide our construction of features for predicting change in rank position. A straightforward myopic decision making step is proposed to address **Q2**. Prediction results are provided in Section 6.5 answering **Q1**. Some insights from the knowledge discovery process are mentioned in Section 6.6 in the attempt to answer **Q3**.

6.2 Related Works

Work on knowledge discovery systems in different domains have highlighted some of the important challenges that we also face in this work [see for instance Fayyad et al., 1996, Frawley et al., 1992, Hand, 1994, Langley and Simon, 1995, Provost and Kohavi, 1998, Brodley and Smyth, 1997, Saitta and Neri, 1998, Rudin and Wagstaff, 2013]. In particular, these works have highlighted the importance of designing knowledge discovery systems around the unique aspects of a domain. These works also emphasize the key choice of proper evaluation metrics, and being able to provide insight that goes beyond prediction accuracy, and back to the important aspects of the domain. The choice of machine learning algorithm itself is not always a critical choice within a knowledge discovery system; in our data mining step, we found that several different algorithms have essentially similar performance.

There have been few recent attempts to use prediction models for in-game decision making in sports such as baseball [Gartheeban and Guttag, 2013, Ganeshapillai and Guttag, 2012], basketball [Bhandari et al., 1997] and cricket [Bailey and Clarke, 2006, Sankaranarayanan et al., 2014]. This is contrast with season level statistical modeling which is well researched in the literature, due to applicability in sports betting and fantasy sports in addition to helping the teams improve their competencies. See Schumaker et al. [2010] for a brief overview. Note that for professional racing, season level research has been sparse [see for instance Graves et al., 2003, Pfitzner and Rishel, 2005, Depken and Mackey, 2009, Allender, 2011] and our work is the first to explore in-race predictive modeling.

For baseball, Gartheeban and Guttag [2013] developed a prediction model to decide when to change the starting pitcher as the game progresses. Similar to our workflow, they proposed several features from historical data and the current game's history to predict a pitcher's performance. At a given point in the game, they forecast the future performance of the pitcher, compare it to a pre-defined threshold and make a binary myopic decision whether the pitcher should continue or not. A related work [Ganeshapillai and Guttag, 2012] looks at predicting the type of pitch that will be thrown by a pitcher given the current state of the game and historical data about the teams playing.

In basketball, Bhandari et al. [1997] developed a knowledge discovery and data mining framework for the NBA (National Basketball Association) with the aim to discover interesting patterns from basketball games. This and related (often proprietary) systems have been in operation with many basketball teams over the past decade. Such solutions are tailored for offline use and do not address in-game prediction and decision making as we do. There has also been some recent work [Skinner, 2012] exploring in-game decision making as a function of time remaining in the game without building any prediction models.

A key difference between predictive modeling for professional racing compared to that in basketball (and baseball) is the nature of the evolution of the game. In racing, the race history cannot be easily segmented into "plays". At each point in time of a race, the entire history of the race determines the racer's current rank position. On the other hand, in basketball, the game is restarted at the beginning of each play, and the team's current state does not heavily depend on their state before the restart. One can reasonably approximate a basketball game to be a sequence of independent plays, and even model them as independent observations drawn from a distribution. These long-standing correlations of decisions within the race makes racing inherently much more difficult to model.

In cricket, Bailey and Clarke [2006] and Sankaranarayanan et al. [2014] explored machine learning methods to predict the future states of the game given features related to the current state of the game and the features of the two teams competing. They consider both season level data and the data collected within the game to predict future scores. Although both these works are closer to what we do, there are a couple of key differences: (a) these works involve a much lower dimensional prediction problem (about 15 features in Sankaranarayanan et al. [2014]) compared to ours ($>$ 100, see Section 6.4), and (b) professional racing involves many more strategic agents (for NASCAR, about 40 racers race) compared to cricket (2 teams, which is is also the case for basketball and baseball). We believe having a high number of strategic agents can have significant impact on predictability and makes the knowledge discovery process more critical compared to two-team games.

Another key feature of our work is that we explore the knowledge discovery pipeline extensively compared to the previous works. This is partially because for basketball, baseball and even cricket, there has been significant prior academic research output compared to professional racing. In this work, we critically examine many details and characteristics of NASCAR in Section 6.3. For instance, we observe Simpson's paradox-like phenomena between two explanatory variables (slope of lap times and number of tires changed). Our exploration of data can help future work on racing focus more on statistical modeling and prediction as in baseball and basketball.

The need for predictions at the finest granularity of racing is two-fold: 1) Previous studies on racing, like those using only race-level and season-level statistics may be too coarse to be beneficial within the middle of a race. For example we believe that statistics computed during the race, for instance, the state of the race after 100 laps, often reveals more about the outcomes of the current race than the predictions made by the previous studies. Season level and multi-year studies are also susceptible to changes in the rules or other changes to the sporting event. For example, for NASCAR, rules have changed multiple times, the latest ones being in 2008 and 2011. This further reduces the effectiveness of race-level statistics for aiding racing *strategy.* 2) By calculating within-race predictions dynamically as the race evolves, we can better quantify the contribution of real-time observations towards predicting outcomes in each portion of the race.

Finally, we note that the approach we take to building a knowledge discovery framework and decision making system for professional racing can be applied to other racing sports with similar structural characteristics, including MotoGP [see also Streja, 2012], Formula 1, IndyCAR, various other types of races within NASCAR, and also bicycle races and marathons.[2]

6.3 Data and observations

We define some of the race-specific terms used in the chapter:

- Lap: One full trip around the race track.

[2] *MotoGP* is a motorbike racing competition where races last about 30-45 minutes with 20-30 laps. *Formula 1* races are quite different than NASCAR races in that the cars within the same race can be mechanically very different, the rules are different, and the level of data can be at a much finer granularity. *IndyCAR* racing is similar to NASCAR racing but the type of car is different. NASCAR has several different stock car and truck races beyond the particular series in our dataset.

- Lap time: The time for a racer to finish one lap.

- Rank position: The position of the racer at the end of a lap. If the position is 1, the racer is leading the race.

- Pit stop: The event in the race when a racer stops racing and enters the pit (area where cars are serviced) with the intention of changing the tires or refueling.

- Caution lap, or yellow lap: A lap is called a caution lap[3] when the racers are not actively racing, have slowed down and are following a "safety car." Caution flags (yellow flags) are displayed due to a hazard on the track (crash, tire burst, etc). In our racing dataset, caution flags are a random influence that substantially affect race dynamics.

- Green lap: Laps which are not in caution are called green laps.

- Warm-up period: After a racer's pit stop or after the end of a caution, the warm up period includes green laps in which the lap times are decreasing successively as the car gains speed.

- Epoch: The green laps after the warm up period until the next pit stop or caution lap constitute an epoch.

- Outing: The green laps in the warm-up period and epoch together form an outing for the racer.

In our study, we use race data constituting 119,178 lap times and 119,178 rank position observations from 2,932 total outings, including each racer's lap times and rank positions for each one of the 5,352 laps within our dataset. We also have caution lap and pit stop information (time, number of tires changed) for each racer. (Some races have unusual race characteristics, for instance some are road courses and some had insufficient or missing tire change information. Thus these were not used in our study.) Races comprising this dataset are listed in Table 6.1. The number of laps in

[3] The rules that define a caution lap are different for different types of professional races. The definition we provide suffices for our analysis of NASCAR races.

Table 6.1: Summary of the 17 race dataset used in our experiments.

NASCAR Sprint Cup 2012 Dataset			
Bristol First	Bristol Second	Charlotte First	Chicago
Darlington	Homestead	Kansas First	Kansas Second
Kentucky	Loudon First	Loudon Second	Martinsville Second
Michigan First	Michigan Second	Phoenix Second	Pocono First
Vegas			

the 17 races we consider ranges between 160 and 500 laps. The total number of pit stops per race varies between 170 and 373 and the average number of pit stops per racer varies from 4 to 8.9. The number of cautions varies between 3 and 14.

6.3.1 Complexities of Racing

To give a sense of the difficulty in modeling with racing data, we next discuss general characteristics of racing and how nonlinear interactions between measurements and other issues pose a difficulty in modeling and decision making. Several of these observations have not (as far as we know) been previously quantified, in particular the “fresh air" effect and the Simpson's paradox effect from tire change decisions discussed below.

Tire change decisions: As we discussed, this is a major strategic decision for each team. In isolation, a car with four fresh tires is generally faster than a car with only two fresh tires, however, it is not that simple during a race: the speed of racers is heavily dependent on more than just tire freshness; as we will discuss, rank position and the ability to overtake other racers plays an important role in determining speed. A two tire change may or may not be an overall advantage depending on whether the racer is also able to maintain their rank position.

Choosing a two tire change saves a racing team about 6 seconds on average over a four tire change, though there is a high variance in pit times. Pit lanes have speed limits that dictate the minimum pit road time, and the racer has to slow down from the speed limit while stopping at his designated stop, make turns into and out of his stop and avoid other racers executing pit stops around him. These elements and

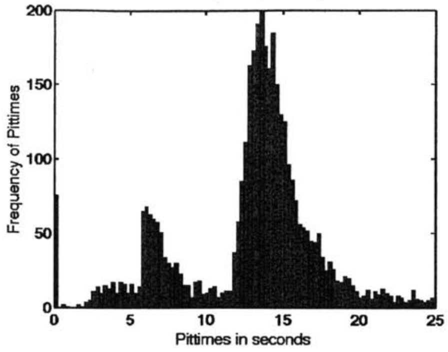
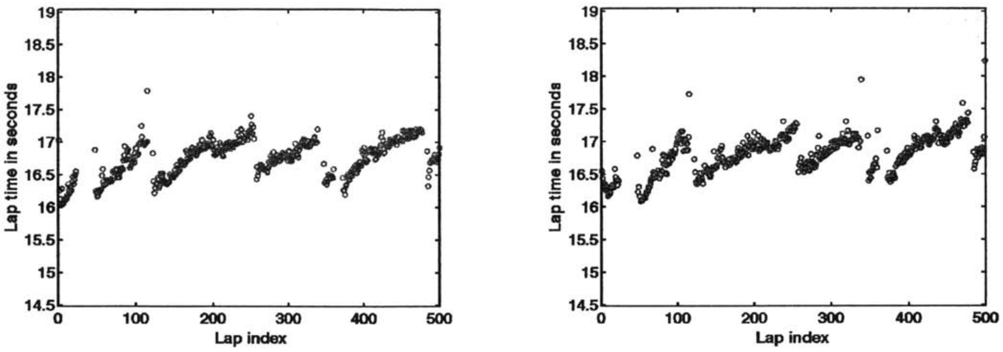


Figure 6-1: Histogram of pit times taken by various racers in our dataset.

the actual performance of the pit crew in servicing the car determine the pit stop times. Figure 6-1 shows the histogram of pit times. One can see three peaks (around 4 seconds, 7 seconds and 14 seconds) and a peak at 0 seconds. The 0 second pit times are due to penalties among other causes (including missing data defaulting to 0). The other three peaks are due to the decision to replace zero, two and four tires respectively. A zero tire pit stop is for refueling only.

Saw tooth profile of lap times: Examples of the lap-time time series for typical racers in our dataset is shown in Figure 6-2. Lap times increase (the car gets slower) as the tires wear down over the course of an outing. Towards the end of an outing, one can also see that the lap times sometimes flatten out; the lap times deteriorate at a slower pace later in the outing. We use the *slope* (estimated rate of change in seconds per lap) of these lap times over the course of an outing to measure tire wear. See Figure 6-3 for an example of how slopes are computed.

The "fresh air" effect, which is a nonlinear interaction between lap time and rank position: In general, lap times are lower (better) for racers near the front of the pack. This is illustrated in Figure 6-4 for three typical laps in three different races. Remarkably, a *linearly* increasing trend is plainly visible between lap time and rank position in each figure. That is, the lap speeds of racers at the front of the pack can be substantially faster than those in the middle of the pack, which can be substantially faster than racers at the back of the pack.



(a) Racer who finished in rank 1

(b) Racer who finished in rank 15

Figure 6-2: Sawtooth profile of typical racers in a race.

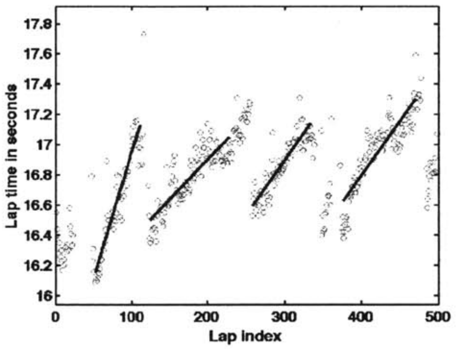
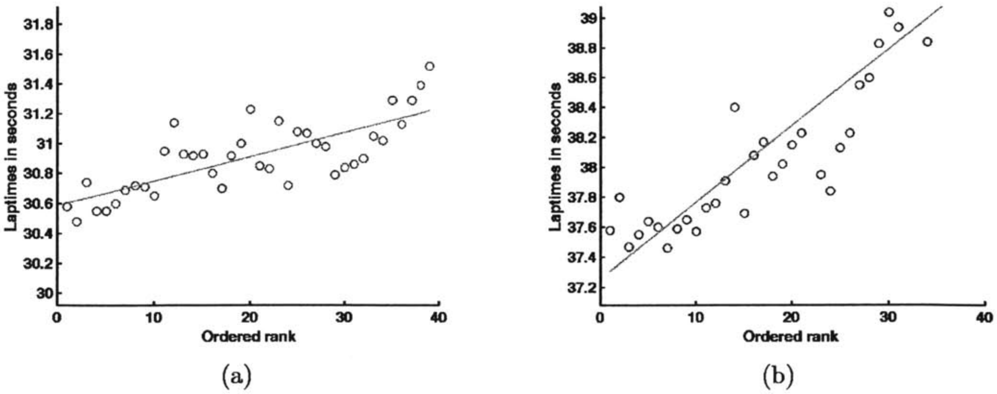


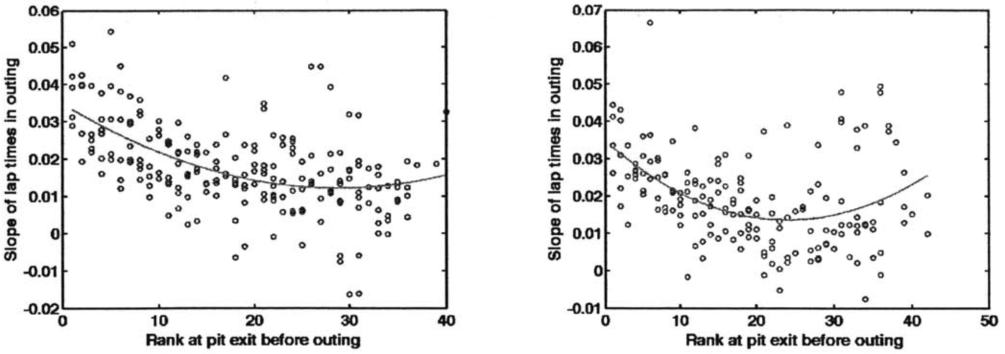
Figure 6-3: Plot of lap times and linear fits for a 15th ranked racer in a race. Slopes are computed by fitting a line through the lap times in an outing using simple linear regression.



(a)

(b)

Figure 6-4: Fresh air effect: ordered lap times of the racers at lap 50, sorted by rank position, for two separate races. Each dot represents a racer's lap time. There are about 40 racers in each plot.



(a) (b)

Figure 6-5: Slopes of lap times within an outing vs initial rank in the outing, for two separate races. Each dot represents a racer's outing within a race. In a typical race, each racer has multiple outings; thus, there are multiple dots for each of the ~ 40 racers in each race.

Because racers near the front of the pack tend to go faster, their tires tend to wear out more quickly. In fact, we observe that the slope of lap times over an outing increases more quickly for cars at the front of the pack. This is shown in Figure 6-5. Actually this effect is highly nonlinear: the cars in the front of the pack and the back of the pack tend to have higher slopes, and the cars in the middle tend to have lower slopes. The effect is fit nicely by a degree-2 polynomial, as shown in Figure 6-5.

Simpson's paradox[4] [Simpson, 1951] for the number of tires changed and the slope: Consider Figure 6-6's leftmost subplot, which shows the distribution of slopes for two tire changes and the distribution of slopes for four tire changes during a race. It is clear that in this race, cars that took two tires had much faster wear (higher slopes) than cars that took four tires. This seems to indicate that older tires tend to wear faster for this race, and thus if the epochs are sufficiently long, it would generally be strategic to take four tires. However this is a severely incomplete picture.

[4] Simpson's paradox occurs when conclusions drawn from parts of a dataset are the opposite of conclusions drawn from the union of these parts. For example, let $\frac{p_{i,j}}{q_{i,j}}$ with $i = 0, 1$ and $j = 0, 1$ be the fractional frequencies of co-occurrence of a factor i and a lurking factor j. Then, a Simpson's like paradox occurs due to the following:

$$\frac{p_{0,0}}{q_{0,0}} > \frac{p_{1,0}}{q_{1,0}} \text{ and } \frac{p_{0,1}}{q_{0,1}} > \frac{p_{1,1}}{q_{1,1}} \text{ does } \textit{not} \text{ imply } \frac{p_{0,0}+p_{0,1}}{q_{0,0}+q_{0,1}} > \frac{p_{1,0}+p_{1,1}}{q_{1,0}+q_{1,1}}.$$

In fact rank position is a lurking variable, in the sense of Simpson's paradox, and has the following effects:

(a) Because only cars that have generally better rank positions take two tires, their slopes are also higher (as we showed in Figure 6-5). In fact, for racers in ranks 26-43, there are no instances of two tire changes compared to 49 instances of four tire changes. This results in a lower median slope for four tire changes, as shown in the leftmost subplot in Figure 6-6.

(b) If we break down our data according to rank positions 1-5, 6-15 and 16-25 as shown in the three subplots to the right in Figure 6-6, we see that the median slope values across ranks are actually very similar for two tire changes and for four tire changes, in seeming contradiction with the leftmost boxplot.

Thus, conclusions drawn from simply looking at slopes for two tire changes and slopes for four tire changes, as in the left of Figure 6-6, would be misleading. Note that the impact of the two or four tire decision depends on many factors besides rank position. When the distribution of slopes are similar as in the box plots for rank positions 1-5, two tire changes would be strategic since the racer could gain rank position without any predictable change in the rate of tire wear.

Race dynamics around a green lap pit stop are different from those after a caution lap pit stop: Racers may choose to pit during a green lap to refresh tires and/or refuel. Not all cars take green lap pit stops around the same time, which causes a high variance in rank positions around the laps when these pit stops occur. For instance, a 20th rank position racer, who has been in the same position through the outing, can become a first rank position racer temporarily if the 19 racers in front of him pit while he does not. Usually, he will then pit in the succeeding laps. While the other cars are in the pit and he is not, his first rank position is artificial. Also, in this case, his pit entry rank position would be recorded as 1. Thus, the green lap pit stops can be very problematic for our analysis, as rank position is not completely meaningful when other racers are in the pits. Caution lap pit stops, on the other hand, are less susceptible to high variability. In the case of outings preceded by green

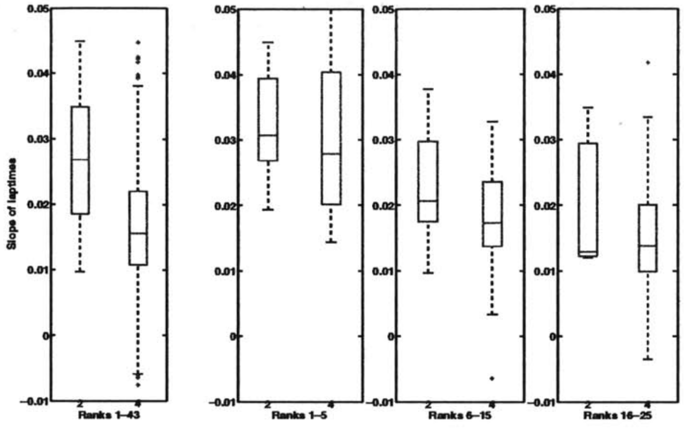


Figure 6-6: On the left, we include slopes for all ranks on a single boxplot. The right three boxplots again show the distribution of slopes, but separated by rank position. Rank position can be considered the lurking variable for Simpson's paradox, as the right three boxplots refute the hypothesis from the left boxplot - namely, that the slopes for two tire changes are substantially larger than the slopes for four tire changes. In these boxplots, there were 26 two-tire changes and 176 four tire changes. These data are from a track in the midwest of the U.S.

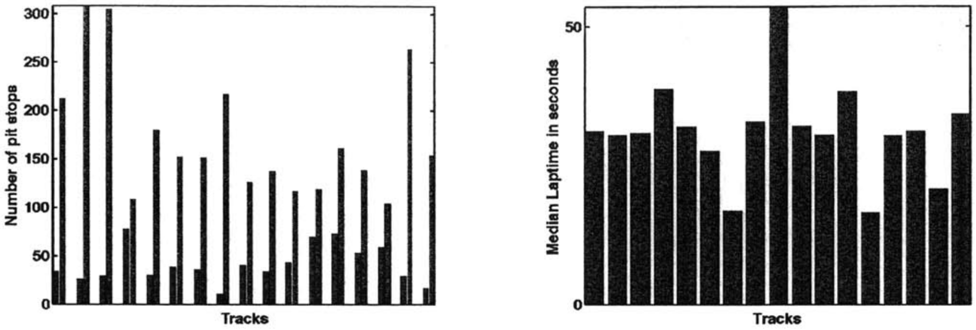
lap pit stops, the racers are more spread out on the track than in the outings preceded by caution lap pit stops (which are similar to a race restart).

Game theoretic aspects (neighborhood interaction): Neighboring racers impact each other due to shared track space. This is a key difference from other racing sports like athletic short distance track events or indoor swimming where there is minimal neighborhood influence since each player has their own assigned lane.

6.3.2 Data issues

Besides the inherent complexities of racing discussed above, there are some natural challenges that arise when making decisions based on historical data. In NASCAR, the decision to replace two tires vs four tires is one such case, particularly due to the data problems of control, imbalance and noise described below.

No controlled experiments: Recall that our objective was to make informed decisions (two tire or four tire) based on race history. Unfortunately, we cannot perform randomized controlled trials in order to measure the effect of a decision; we are lim-



(a) Two and four tire decisions.

(b) Median lap times.

Figure 6-7: Bar plot of two and four tire decisions per race for our dataset is plotted in (a). Left (blue) bars are the total number of two tire decisions in the race and right (red) bars are the total number of four tire decisions. In (b) is a bar plot of median lap times observed per race for our dataset.

ited by what we can do with the historical data. One way to partially handle this shortcoming is to pick "similar" racers who differ only in their tire decisions, and verify whether there is any difference in the causal effect of the decision. Again this is unsatisfactory, as controlling for all other variables in the system is very difficult.

Imbalance: There are far more four tire pit stops than two tire pit stops. This makes it difficult to quantify the effect of the number of tires on the performance of the racer. Figure 6-7(a) shows the number of two and four tire pit stops in each race of our dataset. In addition, almost all practice before a race is based on four tire changes with the intention of tuning the settings of the car. Here, the total number of tires and total laps that can be run are budgeted as well.

Races are different: We would like to be able to generalize knowledge (or borrow strength) across races. However, races can be fundamentally different, prohibiting a straightforward merging of observations across races. The number of laps in the race, the length of the tracks, their physical characteristics (e.g., banking characteristics) can be very different, which all heavily affect lap times. For instance, Figure 6-7(b) shows the median lap times of races we analyze, where the median is taken over all racers and all laps; these heavily vary from race to race. In general, statistics of pit information and lap time information are not race invariant, and cannot be directly compared across races.

Noise: "Irregularities" in racing occur very regularly, such as accidents (hitting the wall, spinning out of control), running completely out of gas, other mechanical failures, and incurring race penalties. These irregularities can affect the quality of our predictions if they are not carefully filtered out. Another aspect that adds to the noise is out-of-sequence pit stops, where a racer takes a pit stop at a different time than the majority, altering the rank positions of others temporarily. Race rules such as "free pass"[5] and strategies such as staying out to lead a lap to earn a point also make our observations noisy.

6.4 Prediction Framework

Keeping in mind the complexities of racing and the data issues discussed above, we now discuss our framework for real-time prediction and strategy in racing.

6.4.1 The prediction problem

Based on Section 6.3.1, we made the following choices about the time scale of learning and the dependent variable.

We chose to forecast the decision-to-decision loss in rank position for each racer, for each decision during the race. This is the change in rank from a car's pit entry to the end of its next outing when it enters the pit again. If we are able to predict this quantity, taking into account the racer's current state, his race history and previous decisions, this will tell us whether the racer's current strategy may give him an advantage between the current decision time and the next one. Note that since a majority of outings end due to cautions, the racer's strategy does not generally determine the end of the outing. The *prediction interval* includes a pit stop and the outing following it, for a given racer. Our system makes a prediction before each prediction interval. Because of this choice of model formulation, our prediction problem

[5] The first of the racers who are one lap down gets to join the racers in the lead lap if a caution occurs.

becomes a supervised learning problem, for which we can use a range of supervised learning techniques.

We chose to model change in rank position and not other functions of the outing (for instance, slope of lap times) because improvement in rank is really the goal of the team, rather than improvements in, for instance, lap time. One might be tempted instead to model the direct results of a tire change decision such as lap times, or equivalently, the slopes. However, slopes of lap times, though indicative of a racer's performance, are not a direct metric of success at the finish of the race. Also, as we discussed earlier, lap time measurements are heavily tied to rank position (see Section 6.3.1). Predicting rank position can still be complicated since, as we discussed earlier, it can depend on the timing of other racers' pit stops.

To build the prediction model, we use all race information from the current racer and his peers up to the pit entry lap index where our prediction interval starts. We also incorporate the team's planned action during the pit while learning from historical data. This naturally leads to the following myopic strategy: given a learned model, we can compute predictions for each planned action (0, 2 or 4 tires) and determine which action(s) might be strategic between now and the next time a decision is made.

6.4.2 Preprocessing

Our model needs to bypass the data issues discussed earlier, for instance the artificial jumps in lap times caused by pit stops and cautions (the jumps in the sawtooth shape of the lap times discussed in Section 6.3.1). The key to this is to correctly create automated definitions of "outings," "warm-up laps," and "epochs." We found that the prediction quality, interpretation of the prediction model, and potential value of predictions to the racers and the teams improved dramatically as a result of improving these model inputs, along with the other preprocessing steps discussed just below. The definition we developed is fairly complicated and not fully discussed. For instance, our definitions are robust to events such as pit stops during green flags which can cause a racer's rank position to be artificially inflated or deflated, impacting results. In the example we gave earlier, a racer with rank position 20 can come into the pit with rank position 1 if the 19 racers in front of him pit before him. To minimize the number of artificially inflated or deflated rank positions in our processed observations, we alter the pit entry lap indices appropriately. This way, the definition of the epoch has a smaller number of laps, and aims to contain only the laps for which cars in front of the racer had not gone into the pit.

6.4.3 Key hypotheses

Based on exploratory analysis of lap time and rank position measurements, we believe the following key hypotheses impact our ability to predict change in rank. To our knowledge, these have not been published before.

"Rank momentum" leads to useful predictive factors: We compute a racer's "rank momentum" based on whether he is generally gaining or losing ranks. Simply, a racer that started at the back of the pack and continues to obtain better rank positions has a different trajectory than a racer that started out at the front of the pack and gradually moves towards the back. Rank momentum may help alleviate issues with the "fresh air" effect described in Section 6.3.1. Rank momentum terms rely on discrete derivatives of rank position time series. They capture information about racers relative to each other. This is different than the slope of lap times ("lap time momentum") which considers the racers in absolute terms, rather than relative to each other.

"Protection" and other neighborhood effects can lead to useful predictive factors: As we discussed, when a racer takes two tires instead of four tires, this can potentially put the racer in a better rank position initially, but he must maintain his position in the outing afterwards to gain ranks. Our evidence suggests that it is sometimes easier for a racer to maintain rank position if several cars behind him also take two tires. This way he is "protected" by the cars behind him - a faster car (for instance one that had taken a four tire change) coming from behind would need to pass several other cars before passing him. Figure 6-8 illustrates this phenomena using race data. Here, in a certain block of the race, the rank profiles for racers who took two tires beforehand are plotted. We see that racers with ranks 13-19 took two

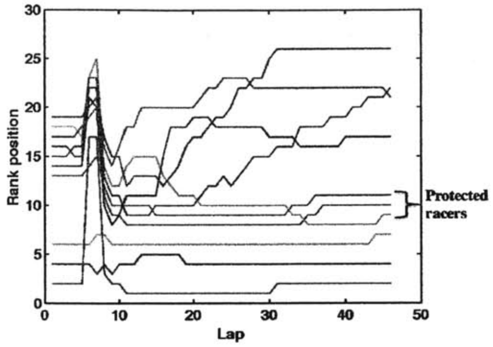


Figure 6-8: An instance of protection: We plot rank position vs. relative lap index for a race. Cars in ranks 2, 4, 6, 13-19 took two tires and the remaining cars took four tires. For clarity, we show only the rank positions of the cars that took two tires during the sixth/seventh lap. The four tire cars overtook some of the two tire cars as seen by the upward moving rank profiles in the upper half of the graph. There were also some two tire cars that did not change rank position as seen by horizontal lines in the lower half of the graph. They were thus *protected* because many of the cars behind them also took two tires.

tires before the outing. About half of these racers maintained their rank position through the outing (see the horizontal lines between ranks 8-11). The remaining half were overtaken by four tire racers behind them (see the upward drifting curves ending between ranks 17-27). We hypothesize that the first group of racers were *protected* from the four tire cars whereas the latter group of cars were not.

There are other possible neighborhood effects besides protection. For instance, we hypothesize that the historical performance of a racer's immediate neighbors can help to predict both change in rank and slope of lap times over the course of an epoch. We considered two types of neighbors: neighbors who hold similar rank positions at the beginning of the current outing's pit exit lap[6], and neighbors who have held similar rank positions and lap times historically within the race (even if they do not hold similar rank positions in the current outing's pit exit lap). These neighborhood effects help to capture correlations across racers, whereas rank momentum captures temporal correlations.

Aggregation across races can be done, and there are two fundamentally

[6] This information needs to be forecasted as it may not be available before the current outing begins.

different types of races. Our evidence suggests that it is possible to generalize across races. That is, we can borrow strength from data of similar races to make improved predictions. This type of across-race regularization helps make the predictive modeling more robust to noise, and helps with the imbalance problem. It is also particularly useful at the start of the race: using another race's data is better than the alternative, which is no data at all.

Through descriptive statistics, we made the hypothesis that there are fundamentally different types of races, namely those for which cars typically lose position after a two tire change (Group A), and those for which cars typically maintain their rank position after a two tire change (Group B). Thus, in Group B, there is more incentive to take two tires instead of four tires to gain rank positions. In reality, the determination of which group the race belongs to can be done using data from practice and qualifying stages that occur on the same track prior to the race. The fact that our observations are *race*-specific rather than *racer*-specific indicate that properties of the track, tires, and weather matter more than racer-specific details in determining how tire change decisions should be made within a race. In our experiments, we did not explicitly use track specific information for this clustering and instead used the given lap position and lap time information to come up with the two groups: Group A (with loss in rank pattern) included six races and Group B (without loss in rank pattern) included the remaining eleven.

6.4.4 Features

Based on the key hypotheses above, we constructed several groups of features for the prediction problem described in Section 6.4.1. These features heavily rely on the definitions and pre-processing we established in Section 6.4.2. We developed over a hundred features, each based on a hypothesis about what might be important for prediction of change in rank over the course of an epoch. The features fall into these categories:

- Basic Features: Basic features are constructed from all the historical outings in the dataset. These are statistics computed from each outing up to the current outing within the current race, and the outings within previous races. Basic features capture: (i) The racer's rank position at the decision time, and whether his rank position is near the top of the pack or near the bottom. We also include the racer's starting rank position for the race. (ii) The average of the racer's rank positions in previous outings (also various percentiles). This indicates how well the racer is doing generally in the race so far. We also include nonlinear variations of this type of feature, such as the average of the previous rank positions squared. (iii) The age of both the left and the right tires at the decision time. (iv) The average of the slopes of the racer's lap times in previous outings, based on fits of each "sawtooth" function. This indicates the general speed of wear of tires for that particular racer. We also use nonlinear functions combining the racer's past rank positions and the average slope, which helps to address the nonlinearity due to "fresh air" discussed above.

- Rank Momentum Features: We compute the minimum, maximum and average of several rank momentum quantities over previous outings within the race. These features include: change in rank, rate of change in rank, change in rank times average rank, and rate of change in rank times average rank.

- Protection Features: We compute statistics of the racer's neighborhood. Here, the neighborhood includes cars within a few ranks of the racer's average rank over the course of the immediately previous outing. These statistics include rank momentum features of the neighborhood. These statistics can help to determine whether the racer might be near cars that he needs to pass, or whether the cars in his neighborhood are likely to be faster than he is, in which case he might lose ranks. We further consider the number of neighbors with zero, two or four tire changes before their outings began.

- Tire Decision Features: The tire decision that happens before the outing is a critical feature whose impact on the change in rank can help us make decisions during the race. We can make product features from tire decision features and other features, like whether the racer has taken two tires and is at the front of the pack, in "fresh air."

- Other Features: These are features that are potentially important, but do not fall into the earlier categories. These features include:
  - An indicator of first outing in the race: The first outing does not have historical information about past outings of the racer. This makes that outing different from all subsequent outings of the race.
  - An indicator of pit in caution: This feature allows us to address green lap pit stops differently than pit stops during cautions.
  - Time taken in previous pit stops: This feature addresses the variability in pit times discussed in Section 6.3.1.
  - An indicator variable for whether the previous outing was short: If the previous outing was very short, it may affect the race dynamics in the current outing. Many racers will not change tires if they have done so recently.

Using these features to aggregate information across races assists with the data issues from Section 6.3.2, specifically imbalance and the lack of information at the start of a race. It is not true, however, that any past race is able to assist with prediction in any current race: our grouping of tracks alleviates this problem.

6.4.5 Prediction to Decision

We built a real-time prediction system by re-solving the batch learning problem at each lap. Specifically, to do this for a given racer, at each lap we compute his predicted change in rank position in the next outing given a zero, two, or four tire decision that he may choose to take in a pitstop in the near future. Comparing these three predicted change-in-ranks against one another helps the crew chief of the team make a well-informed call.

6.5 Experiments

We experimented with several state-of-the art experimental machine learning techniques that permit different combinations of the features we created. In particular, we used ridge regression[7] [Hoerl and Kennard, 1970], support vector regression (SVR)[8] [Drucker et al., 1997] with a linear kernel, LASSO[9] (Least Absolute Shrinkage and Selection Operator) [Tibshirani, 1996] as well as Random Forests for regression [Breiman, 2001a] and two baselines. Ridge regression and LASSO are very similar techniques in that both use the same least squares loss function, but LASSO uses ℓ_1 regularization to determine the coefficients, whereas ridge regression uses ℓ_2 regularization. Support vector regression also uses ℓ_2 regularization, but uses the $\epsilon-$insensitive loss function. Random Forests is an ensemble method that averages predictions from many different decision trees. The two baselines are as follows:

- **Baseline initial rank**: We always predict that the change in rank over the course of the prediction period is zero.

- **Baseline regression to the mean**: We always predict that the final rank at the end of the prediction period will be the racer's average rank from his previous epochs. This means the predicted change in rank will be the difference between his historical average rank and his rank at the beginning of the prediction period.

Because we do not have control over data generation as discussed in Section 6.3.2, the linear model coefficients (e.g., of support vector regression, ridge regression and LASSO) cannot be reliably interpreted in the *ceteris paribus structural form.* This

[7] Given data $\{x_i, y_i\}_{i=1}^n$ and a constant C, we obtain linear model $w^* \in \arg\min_w \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^n (w^T x_i - y_i)^2$.

[8] Similar to ridge regression, we get w^* from solving the following for a fixed parameter $\epsilon > 0$:

$$\begin{aligned} \min_{w,\xi,\xi^*} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \\ \text{subject to} \quad & y_i - w^T x_i \leq \epsilon + \xi_i \ \ \forall i = 1, ..., n \\ & w^T x_i - y_i \leq \epsilon + \xi_i^* \ \ \forall i = 1, ..., n \\ & \xi_i \geq 0, \xi_i^* \geq 0 \ \ \forall i = 1, ..., n. \end{aligned}$$

[9] Similar to ridge regression, we get $w^* \in \arg\min_w \|w\|_1 + C\sum_{i=1}^n (w^T x_i - y_i)^2$.

means that if we are to quantify the effect of the tire decision feature on the subsequent change in rank position, we need the other features to be as orthogonal to the tire decision feature as possible. Nonetheless, our approach is reasonable as prediction performance is also primarily desired.

6.5.1 Metrics

There are no agreed upon domain specific measures of success to employ for our prediction step. We decided to use R^2 (r-squared)[10], RMSE (root mean squared error) and sign accuracy[11] as the evaluation metrics for the prediction models on out-of-sample data. R^2 describes the proportion of variance of the dependent variable (change in rank position) explained by the regressors (features in Section 6.4.4) through the prediction model. For a perfect relationship it is 1 and for no relationship it is 0. Sign accuracy captures the proportion of time we predict correctly whether the rank increased, decreased, or stayed the same.

6.5.2 Prediction performance

We performed two sets of experiments, using data from all outings that were sufficiently long. The first involves predictive accuracy of the different models. In the second experiment, we observe how the weight of the two tire indicator feature changes with outing length.

- **Predictive Accuracy:** We built prediction models for each group. This allows us to investigate the change in prediction performance due to grouping. We adopted the following data splitting strategy for evaluating predictive accuracy: we used the outings at the beginning part of the race in our training and validation sets and reserved the ending part of the race for testing. In this way, we avoid data leakage by training only on the earlier parts of the race to

[10] R^2 is defined to be $1 - \frac{\sum_{i=1}^{n}(y_i - f(x_i))^2}{\sum_{i=1}^{n}(y_i - \frac{1}{n}\sum_{i=1}^{n} y_i)^2}$, where f is the prediction model. Note that R^2 can be positive or negative.

[11] We define sign accuracy to be equal to $\frac{1}{n}\left(\sum_{y_i<0} \mathbf{1}_{[f(x_i)<0]} + \sum_{y_i=0} \mathbf{1}_{[f(x_i)=0]} + \sum_{y_i>0} \mathbf{1}_{[f(x_i)>0]}\right)$.

evaluate predictions for the later parts. We could have also chosen to use all outings of some races in the training and all outings of the rest of the races for final testing. In our experiments, we did not find a noticeable difference using this type of data splitting.

- **Variation of the weight of the two tire decision feature with outing length:** We built prediction models to forecast the change in rank over the current outing at pre-specified laps, namely, one lap after pit exit, two laps after pit exit, and so on up to twenty-five laps after pit exit. Through this experiment, we expect to gain insight on the effect of outing length on feature weights in a linear model like LASSO.

For both of these experiments, we used 5-fold cross validation to set the appropriate regularization coefficient (or parameter values in case of Random Forests). We repeated splitting the data into 5 folds, 10 times to make the cross validation procedure more stable[12] and used the same set of folds for all the models used (to control for split variance).

The results of the first experiment characterizing performance of the methods on test data using different metrics are plotted in Figure 6-9. Figure 6-10 shows the values of the regularization parameters chosen for each group. The results for the second experiment characterizing the effect of outing length on the model weight of the two tire change feature are plotted in Figure 6-11. We summarize some of the findings from these experiments below:

Predictive Accuracy:

- From the prediction performance plots in Figure 6-9, we can see that the ridge regression, SVR, LASSO and Random Forests are significantly better than the baseline methods. The machine learning methods give very similar held out test set performance. Further reduction in RMSE, increase in R^2 and increase in

[12] Since the number of observations is comparable to the number of features, a single 5 fold split may lead to some folds having much less training error than others. For instance, if we split again, we may end up picking a different regularization parameter. We found 10 repeats to give us a cross validation matrix with significantly less variation across folds.

sign accuracy may not be possible because of the highly strategic and dynamic nature of racing.

- Predictions on the test set are somewhat worse than performance on the training set. This is not due to over fitting, it is because the training distribution differs from the test distribution due to the following:

  1. Later outings of a race have different dynamics than the beginning part of the race. For instance, the racers are closer to the finish line in the later outings, so their risk profiles change, leading to more aggressive driving, and typically there are a higher number of cautions.
  2. Two-tire decisions acquire relatively more significance during later outings and are typically observed more during that period of the race. If there are fewer two-tire changes in the earlier part of the race than in the later part, we may not be able to accurately characterize the later part of the race from the earlier part.

Variation of the weight of the two tire decision feature with outing length: In Figure 6-11, we see that in Group A (with loss in rank pattern), there is a positive weight on the two tire change indicator. In Group B (without loss in rank pattern), there is a negative weight on the two tire change indicator. This effect becomes more extreme as the outing length increases. This really shows the difference between the two groups: the effect of a two-tire change can be quite different.

6.6 Some Insights

In this section we highlight some insights and some cases where predictive modeling is able to forecast large change in ranks using the historical features.

Predicting outing length is not critical: We find in our experiments that the length of the outing is not an important predictor of change in rank position as long as it is sufficiently long. This is actually quite useful to know as it saves us the trouble of having to forecast outing length, which is very difficult. The reason for outing length

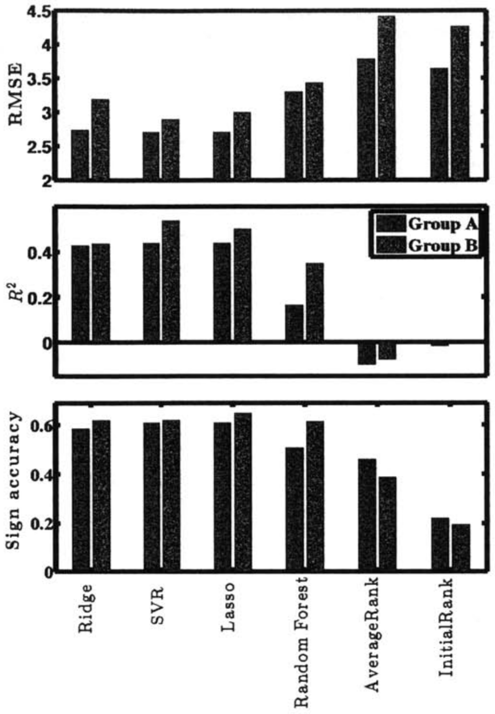
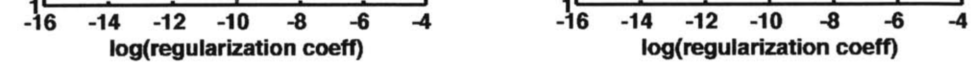


Figure 6-9: Predictive performance of various models over a held out test set are shown for races in Group A and Group B. The y axis plots the RMSE (lower is better) for the top subplot, R^2 (higher is better) for the middle subplot and the sign accuracy (higher is better) for the bottom subplot.



(a) Group A

(b) Group B

Figure 6-10: For both groups of races, we plot the mean (over 10 repeated choices of 5 validation sets) of the mean squared error along with error bands corresponding to 1 standard deviation above and below while building a LASSO model. The vertical line represents the regularization constant for which the mean cross-validation error is the minimum.

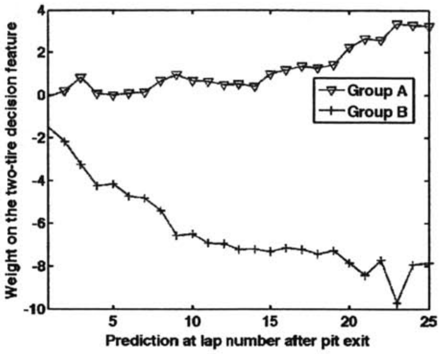


Figure 6-11: Variation in the weight of the two-tire decision feature in LASSO as a function of the outing length. For Group A, the weight is positive and increasing, indicating that making a two tire decision increases the change in rank (loss in rank). This effect increases as the outing length increases. An opposite effect is observed in Group B.

not to be necessary could be that after the initial few laps of a long outing, the racers are typically sufficiently spaced apart on the race track, so that the change in rank position remains relatively constant irrespective of the length of the outing.

Note that this observation does not conflict with (and can actually be seen using) Figure 6-11: as the length of outings increases (towards the right of the figure), the weights stabilize.

It is hard to beat the baseline initial rank with respect to the RMSE: In many of the outings observed, racers typically change their position by zero, one or two ranks. Thus the baseline trivial model that predicts zero change in rank *all the time* does fairly well with respect to the RMSE. It does not, however, perform well with respect to the R^2 or sign accuracy metrics. In fact, since it always predicts zero, and cars stay in the same rank position 20% of the time, the sign accuracy is 20%.

Validation through expert commentary: Expert commentaries[13] that are typically stated either before or after the race can also be used to qualitatively validate the inferences of our modeling approach. For example, some commentaries about the characteristics of tracks that influence racing strategies and outcomes for 2012 were:

- "As your fuel load burns off, you gain a little bit of speed on track... the tires

[13] For instance, based on pre-race comments by the crew chief of car 48 for 2012, among others.

aren't falling off much ..."

- "I don't think tire wear is going to be very high ..."
- "Tires don't really seem to be making a huge difference in lap times ..."
- "... crew chiefs must decide whether to pit or not and whether to take two tires or four."
- "... you are going to see two tires, you are going to see four tires ..."

When we looked at the tracks that the experts were commenting on, we found that the first three comments corresponded to tracks in Group B. Recall that Group B are tracks for which the number of tires changed tends not to matter, and where we recommended taking two tires rather than four because there is no loss in rank pattern. Our grouping agreed with the expert commentary in all three cases. The last two comments corresponded to tracks in Group A, where we correctly identified that there was a perceivable effect of a two tire strategy on rank position outcomes.

There are other types of commentaries that are useful in decision making but are not directly related to our grouping. For instance, some tracks have far spaced and few caution lap periods. This is because the track is wide, which reduces the possibility of cautions, and in turn affects the tire strategy of racers. Thus these commentaries also help to justify our clustering of races before fitting the prediction models.

Insights for some extreme outings observed in the dataset: It is of particular interest to the teams to understand outings where a high change in rank occurs. We now present some representative cases where change in rank was significantly high and moderately predictable. See Table 6.2 for a numerical summary of these cases. We qualitatively describe why our prediction model (in particular, LASSO) was able to predict in these 'high' change in rank cases. LASSO outputs a linear model, that is, it provides a weight for each feature, and the weighted sum of features is the predicted change in rank. These weights can be positive or negative.

Fifth outing for car #5 in a race in the southern U.S. : Our model pinpointed two main reasons why this particular racer should gain ranks in the next epoch: this racer was towards the back of the pack, and his tires did not wear out as quickly as the other racers in the previous epoch (as indicated by the slope of his lap times). To show how our model does this, we note first that the feature *rank(pit entry lap)* encodes that his rank is towards the back. Second, we note that the feature *slope(laptimes of previous outing)*× *rank(pit exit lap)* incorporates the fact that his tires did not wear out as quickly as usual for someone in his rank through a low slope in lap times. Further this race is in Group B, which means that two tire changes do not cause as many losses in rank position. As it turns out, in this epoch, the racer took two tires, we predicted that with this choice he would gain a large number of rank positions (10.36), and he gained an even larger number of rank positions (17).

Fifth outing for car #31 in a race in the southern U.S. : This racer was near the front of the pack, and in the previous outing, his slope was relatively high for his rank, indicating that his tires were wearing out more quickly than other racers. Because of this, again our model used the features *rank(pit entry lap)* and *slope(laptimes of previous outing)*× *rank(pit exit lap)* to predict that he would lose a lot of ranks over the next outing. He took zero tires, and we predicted that he would lose 6.11 ranks, and he lost 13 ranks.

Fifth outing for car #2 in a race in the northern U.S. : Similar to the previous case, this racer was near the front of the pack through most of the race. But in contrast, his slope was relatively low for his rank in the previous outing indicating that he had a fast car or his tires were wearing out slower than other racers. In particular, our model used the most dominating feature *slope(laptimes of previous outing)*× *rank(pit exit lap)* to predict that he would gain ranks over the next outing. He took two tires, and we predicted a gain of 2.88 ranks whereas in reality, he gained 5 ranks.

Eighth outing for car #29 in a race in the southern U.S. : This racer alternated between being near the front of pack and being near the back of the pack in his previous outings. His rank was low at pit entry for the outing of interest here. In addition, in the immediate previous outing, his lap times had a high slope (indicating a slower car or relatively more tire wear). Our model used the features *rank(pit entry lap)* and *slope(laptimes of previous outing)*× *average-rank(previous outing)* to predict that he would lose ranks over the next outing. We predicted a loss of 3.77 ranks and the ground truth was that he lost 10 ranks (and took 2 tires before the outing).

In all the above cases, many other features were also influencing the change (loss) in rank variable including features related to the past two tire and four tire changes, *slope(laptimes of previous outing)*×*final-rank(previous outing)*, functions like square root and square of *final-rank(previous outing)* among others. Their influence was relatively smaller for these outings.

Table 6.2: Some extreme cases where the change in rank variable is high and our prediction models are able to predict moderately well. Negative change in rank values mean that the racer gained positions by the end of the outing compared to the pit entry before the outing. All the outings here are towards the end of the race.

Car #	Outing number	True change in rank	Predicted change in rank	Tire decision
5	5	-17	-10.36	2
31	5	13	6.11	0
2	5	-5	-2.88	2
29	8	10	3.77	2

6.7 Conclusion

We described challenges in formulating a prediction problem that leads into the design of decision making tools for strategic use within a professional sporting event. Careful use of domain knowledge and transformation of time series data into a supervised learning framework were the key aspects in our ability to do this. We demonstrated the validity of our prediction models using data from a professional NASCAR racing season in 2012.

Bibliography

Shivani Agarwal. Ranking on graph data. In *Proceedings of the 23rd International Conference on Machine Learning*, 2006.

Sivan Aldor-Noiman, Paul D. Feigin, and Avishai Mandelbaum. Workload forecasting for a call center: Methodology and a case study. *The Annals of Applied Statistics*, 3(4):1403–1447, 2009.

Mary Allender. Predicting the outcome of NASCAR races: The role of driver experience. *Journal of Business & Economics Research (JBER)*, 6(3), 2011.

Theodore Wilbur Anderson. *An introduction to multivariate statistical analysis*, volume 2. Wiley New York, 1958.

Martin Anthony and Peter L. Bartlett. *Neural network Learning: Theoretical Foundations.* Cambridge University Press, 1999.

Aaron Archer and Anna Blasiak. Improved approximation algorithms for the minimum latency problem via prize-collecting strolls. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 429–447, 2010.

Aaron Archer, Asaf Levin, and David P. Williamson. A faster, better approximation algorithm for the minimum latency problem. *SIAM J. Comput.*, 37(5):1472–1498, 2008.

Sanjeev Arora and George Karakostas. A $2 + \epsilon$ approximation algorithm for the k-MST problem. *Math. Program.*, 107(3):491–504, 2006.

Kevin Bache and Moshe Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

Michael Bailey and Stephen R Clarke. Predicting the match outcome in one day international cricket matches, while the game is in progress. *Journal of sports science & medicine*, 5(4):480, 2006.

M.F. Balcan and A. Blum. A PAC-style model for learning from labeled and unlabeled data. In *Proceedings of Conference on Learning Theory*, pages 69–77. Springer, 2005.

Fran Barbera, Helmut Schneider, and Peter Kelle. A condition based maintenance model with exponential failures and fixed inspection intervals. *The Journal of the Operational Research Society*, 47(8):pp. 1037–1045, 1996.

Andrew R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *Information Theory, IEEE Transactions on*, 39(3):930–945, 1993.

Peter L. Bartlett and Shahar Mendelson. Gaussian and Rademacher complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3: 463–482, 2002.

Peter L Bartlett, Philip M Long, and Robert C Williamson. Fat-shattering and the learnability of real-valued functions. *Journal of Computer and System Sciences*, 52 (3):434–452, 1996.

Peter L Bartlett, Olivier Bousquet, and Shahar Mendelson. Localized rademacher complexities. In *Computational Learning Theory*, pages 44–58. Springer, 2002.

Russell R. Barton, Barry L. Nelson, and Wei Xie. A framework for input uncertainty analysis. In *Winter Simulation Conference*, pages 1189–1198. WSC, 2010.

Sugato Basu, Mikhail Bilenko, Arindam Banerjee, and Raymond J Mooney. Probabilistic semi-supervised clustering with constraints. In *Semi-supervised learning*, pages 71–98. Cambridge, MA. MIT Press, 2006.

M. Belkin and P. Niyogi. Semi-supervised learning on riemannian manifolds. *Machine Learning*, 56(1):209–239, 2004.

Mikhail Belkin, Irina Matveeva, and Partha Niyogi. Regularization and semi-supervised learning on large graphs. In *Proceedings of Conference on Learning Theory*, pages 624–638. Springer, 2004.

Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi S. Nemirovskii. *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press, 2009.

D. Bertsimas, V. Gupta, and N. Kallus. Data-driven robust optimization. *Technical report*, 2013. URL `http://arxiv.org/abs/1401.0212`. Submited to Operations Research.

Inderpal Bhandari, Edward Colet, Jennifer Parker, Zachary Pines, Rajiv Pratap, and Krishnakumar Ramanujam. Advanced scout: Data mining and knowledge discovery in nba data. *Data Mining and Knowledge Discovery*, 1(1):121–125, 1997.

John R. Birge and François Louveaux. *Introduction to Stochastic Programming*. Springer Verlag, 1997.

Avrim Blum, Prasad Chalasani, Don Coppersmith, Bill Pulleyblank, Prabhakar Raghavan, and Madhu Sudan. On the minimum latency problem. *ArXiv Mathematics e-prints*, September 1994.

Pierre Bonami, Lorenz T. Biegler, Andrew R. Conn, Gérard Cornuéjols, Ignacio E. Grossmann, Carl D. Laird, Jon Lee, Andrea Lodi, François Margot, Nicolas W. Sawaya, and Andreas Wächter. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5(2):186–204, 2008.

Olivier Bousquet. New approaches to statistical learning theory. *Annals of the Institute of Statistical Mathematics*, 55(2):371–389, 2003.

Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001a.

Leo Breiman. Statistical modeling: The two cultures. *Statistical Science*, 16(3): 199–231, 2001b.

Carla Brodley and Padhraic Smyth. Applying classification algorithms in practice. *Statistics and Computing*, 7:45–56, 1997.

Lawrence D. Brown, Ren Zhang, and Linda Zhao. Root-unroot methods for non-parametric density estimation and poisson random-effects models. *Department of Statistics University of Pennsylvania, Tech. Rep*, 2001.

Giuseppe Calafiore and Marco C Campi. Uncertain convex programs: randomized solutions and confidence levels. *Mathematical Programming*, 102(1):25–46, 2005.

Venkat Chandrasekaran, Benjamin Recht, Pablo A Parrilo, and Alan S Willsky. The convex geometry of linear inverse problems. *Foundations of Computational Mathematics*, 12(6):805–849, 2012.

M Chang, Lev Ratinov, and Dan Roth. Constraints as prior knowledge. In *ICML Workshop on Prior Knowledge for Text and Language Processing*, pages 32–39, 2008a.

Ming-Wei Chang, Lev-Arie Ratinov, Nicholas Rizzolo, and Dan Roth. Learning and inference with constraints. In *AAAI Conference on Artificial Intelligence*, pages 1513–1518, 2008b.

Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

Abraham Charnes and William W Cooper. Chance-constrained programming. *Management Science*, 6(1):73–79, 1959.

Nello Cristianini and John Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

Imre Csiszár and G. Tusnády. Information geometry and alternating minimization procedures. *Statistics and Decisions*, 1(Suppl.):205–237, 1984.

Felipe Cucker and Steve Smale. On the mathematical foundations of learning. *Bulletin-American Mathematical Society*, 39(1):1–50, 2002.

Jeśus A. De Loera. The many aspects of counting lattice points in polytopes. *Mathematische Semesterberichte*, 52(2):175–195, 2005.

Erick Delage and Yinyu Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research*, 58(3): 595–612, 2010.

Craig Depken and Larisa Mackey. Driver success in the NASCAR Sprint Cup Series: The impact of multi-car teams. *Available at SSRN 1442015*, 2009.

Harris Drucker, Chris JC Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. Support vector regression machines. *Proceedings of the Neural Information Processing Systems*, pages 155–161, 1997.

C. A. Eijl van. A polyhedral approach to the delivery man problem. Technical report, Memorandum COSOR 95–19, Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands, 1995.

Şeyda Ertekin, Cynthia Rudin, and Tyler McCormick. Predicting power failures with reactive point processes. In *Proceedings of AAAI Late Breaking Track*, 2013.

Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, 17:37–54, 1996.

Matteo Fischetti, Gilbert Laporte, and Silvano Martello. The delivery man problem and cumulative matroids. *Oper. Res.*, 41:1055–1064, November 1993.

William J. Frawley, Gregory Piatetsky-Shapiro, and Christopher J. Matheus. Knowledge discovery in databases: an overview. *AI Mag.*, 13(3):57–70, 1992.

Simon French. *Decision Theory: An Introduction to the Mathematics of Rationality.* Halsted Press, 1986.

Peter A Frost and James E Savarino. An empirical bayes approach to efficient portfolio selection. *Journal of Financial and Quantitative Analysis*, 21(3):293–305, 1986.

Glenn M Fung, Olvi L Mangasarian, and Jude W Shavlik. Knowledge-based support vector machine classifiers. In *Proceedings of Neural Information Processing Systems*, pages 521–528, 2002.

Gartheeban Ganeshapillai and John Guttag. Predicting the next pitch. In *Sloan Sports Analytics Conference*, 2012.

Ganeshapillai Gartheeban and John Guttag. A data-driven method for in-game decision making in mlb: when to pull a starting pitcher. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 973–979. ACM, 2013.

Pascal Germain, Alexandre Lacasse, François Laviolette, and Mario Marchand. PAC-Bayesian learning of linear classifiers. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 353–360. ACM, 2009.

Michel Goemans and Jon Kleinberg. An improved approximation ratio for the minimum latency problem. *Mathematical Programming*, 82:111–124, 1998.

Donald Goldfarb and Garud Iyengar. Robust portfolio selection problems. *Mathematics of Operations Research*, 28(1):1–38, 2003.

Luis Gómez-Chova, Gustavo Camps-Valls, Jordi Munoz-Mari, and Javier Calpe. Semisupervised image classification with laplacian support vector machines. *Geoscience and Remote Sensing Letters, IEEE*, 5(3):336–340, 2008.

Todd Graves, C Shane Reese, and Mark Fitzgerald. Hierarchical models for permutations: Analysis of auto racing results. *Journal of the American Statistical Association*, 98(462):282–291, 2003.

David J. Hand. Deconstructing statistical questions. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, 157(3):317–356, 1994.

Sven Ove Hansson. *Decision Theory: A Brief Introduction.* Online manuscript. Department of Philosophy and the History of Technology, Royal Institute of Technology, Stockholm, 1994.

Aiwina Heng, Andy C.C. Tan, Joseph Mathew, Neil Montgomery, Dragan Banjevic, and Andrew K.S. Jardine. Intelligent condition-based prediction of machinery reliability. *Mechanical Systems and Signal Processing*, 23(5):1600 – 1614, 2009.

Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

Waltraud Huyer and Arnold Neumaier. Global optimization by multilevel coordinate search. *J. of Global Optimization*, 14:331–355, June 1999.

G. M James, C Paulson, and P Rusmevichientong. The constrained lasso. *working paper*, 2014.

Yaochu Jin. *Multi-Objective Machine Learning, In Studies in Computational Intelligence*, volume 16. Springer, 2006.

Fritz John. Extremum problems with inequalities as subsidiary conditions. *Studies and Essays Presented to R. Courant on his 60th Birthday, January 8, 1948*, pages 187–204, 1948.

Lee K. Jones. A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training. *The Annals of Statistics*, 20(1):608–613, 1992.

Philippe Jorion. Bayes-Stein estimation for portfolio analysis. *Journal of Financial and Quantitative Analysis*, 21(3):279–292, 1986.

Matti Kääriäinen. Generalization error bounds using unlabeled data. In *Proceedings of Conference on Learning Theory*, pages 127–142. Springer, 2005.

W. Kahan. Circumscribing an ellipsoid about the intersection of two ellipsoids. *Canadian Mathematical Bulletin*, 11(3):437–441, 1968.

S.M. Kakade, K. Sridharan, and A. Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. *Proceedings of Neural Information Processing Systems*, 22, 2008.

Roger Koenker. *Quantile regression. Econometric society monograph series.* Cambridge University Press, 2005.

Andrey Nikolaevich Kolmogorov and Vladimir Mikhailovich Tikhomirov. ε-entropy and ε-capacity of sets in function spaces. *Uspekhi Matematicheskikh Nauk*, 14(2): 3–86, 1959.

Vladimir Koltchinskii and Dmitriy Panchenko. Complexities of convex combinations and bounding the generalization error in classification. *The Annals of Statistics*, 33(4):1455–1496, 2005.

Hiroshi Konno and Hiroaki Yamazaki. Mean-absolute deviation portfolio optimization model and its applications to Tokyo stock market. *Management Science*, pages 519–531, 1991.

Agop Koulakezian, Hazem M. Soliman, Tang Tang, and Alberto Leon-Garcia. Robust traffic assignment in transportation networks using network criticality. In *Proceedings of 2012 IEEE 76th Vehicular Technology Conference*, 2012.

Gert RG Lanckriet, Laurent El Ghaoui, Chiranjib Bhattacharyya, and Michael I Jordan. A robust minimax approach to classification. *The Journal of Machine Learning Research*, 3:555–582, 2003.

Pat Langley and Herbert A. Simon. Applications of machine learning and rule induction. *Commun. ACM*, 38(11):54–64, November 1995. ISSN 0001-0782.

Fabien Lauer and Gérard Bloch. Incorporating prior knowledge in support vector machines for classification: A review. *Neurocomputing*, 71(7):1578–1594, 2008.

Quoc V Le, Alex J Smola, and Thomas Gärtner. Simpler knowledge-based support vector machines. In *Proceedings of the 23rd international conference on Machine learning*, pages 521–528. ACM, 2006.

Miriam Lechmann. The traveling repairman problem - an overview. *Diplomarbeit, Universitat Wein*, pages 1–79, 2009.

Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: isoperimetry and processes*, volume 23. Springer, 1991.

Shengqiao Li. Concise Formulas for the Area and Volume of a Hyperspherical Cap. *Asian Journal of Mathematics & Statistics*, 4(1):66–70, 2011.

Miguel Sousa Lobo, Lieven Vandenberghe, Stephen Boyd, and Hervé Lebret. Applications of second-order cone programming. *Linear algebra and its applications*, 284 (1):193–228, 1998.

George G. Lorentz. Metric entropy and approximation. *Bull. Am. Math. Soc.*, 72: 903–937, 1966.

Zhengdong Lu and Todd K Leen. Semi-supervised learning with penalized probabilistic clustering. In *Proceedings of Neural Information Processing Systems*, pages 849–856, 2004.

Marzio Marseguerra, Enrico Zio, and Luca Podofillini. Condition-based maintenance optimization by means of genetic algorithms and monte carlo simulation. *Reliability Engineering & System Safety*, 77(2):151 – 165, 2002.

Andreas Maurer. The Rademacher complexity of linear transformation classes. In *Proceedings of Conference on Learning Theory*, pages 65–78. Springer, 2006.

David A McAllester. PAC-Bayesian model averaging. In *Proceedings of the 12th Annual Conference on Computational Learning Theory*, pages 164–170. ACM, 1999.

Colin McDiarmid. On the method of bounded differences. *Surveys in combinatorics*, 141(1):148–188, 1989.

Isabel Méndez-Díaz, Paula Zabala, and Abilio Lucena. A new formulation for the traveling deliveryman problem. *Discrete Applied Mathematics*, 156(17):3223–3237, 2008.

S. Muthukrishnan, Martin Pal, and Zoya Svitkina. Stochastic models for budget optimization in search-based advertising. *Internet and Network Economics*, pages 131–142, 2007.

John Ashworth Nelder and Roger Mead. A simplex method for function minimization. *Computer Journal*, 7(4):308–313, 1965.

Nam Nguyen and Rich Caruana. Classification with partial labels. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 551–559. ACM, 2008a.

Nam Nguyen and Rich Caruana. Improving classification with pairwise constraints: a margin-based approach. In *Machine Learning and Knowledge Discovery in Databases*, pages 113–124. Springer, 2008b.

Barry Pfitzner and Tracy Rishel. Do reliable predictors exist for the outcomes of NASCAR races. *The Sport Journal*, 8(2), 2005.

Gilles Pisier. *The volume of convex bodies and Banach space geometry*, volume 94. Cambridge University Press, Cambridge, 1989.

David Pollard. *Convergence of Stochastic Processes.* Springer, 1984.

Foster Provost and Ron Kohavi. Guest editor's introduction: On applied research in machine learning. *Machine Learning*, 30:127–132, 1998.

Philippe Rigollet. Generalization error bounds in semi-supervised classification under the cluster assumption. *Journal of Machine Learning Research*, 8:1369–1392, 2007.

Luis Miguel Rios. Algorithms for derivative-free optimization. *PhD thesis, University of Illinois at Urbana-Champaign*, pages 1–133, 2009.

Cynthia Rudin and Robert E. Schapire. Margin-based ranking and an equivalence between AdaBoost and RankBoost. *The Journal of Machine Learning Research*, 10:2193–2232, 2009.

Cynthia Rudin and Gah-Yi Vahn. The big data newsvendor: Practical insights from machine learning. *working paper*, 2014.

Cynthia Rudin and Kiri L. Wagstaff. Machine Learning for Science and Society. *Machine Learning*, To Appear, 2013.

Cynthia Rudin, Rebecca Passonneau, Axinia Radeva, Haimonti Dutta, Steve Ierome, and Delfina Isaac. A process for predicting manhole events in Manhattan. *Machine Learning*, 80:1–31, 2010.

Cynthia Rudin, Rebecca Passonneau, Axinia Radeva, Steve Ierome, and Delfina Isaac. 21st-century data miners meet 19th-century electrical cables. *IEEE Computer*, 44 (6):103–105, June 2011.

Cynthia Rudin, David Waltz, Roger N. Anderson, Albert Boulanger, Ansaf Salleb-Aouissi, Maggie Chow, Haimonti Dutta, Philip Gross, Bert Huang, Steve Ierome, Delfina Isaac, Arthur Kressner, Rebecca J. Passonneau, Axinia Radeva, and Leon Wu. Machine learning for the New York City power grid. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(2):328–345, February 2012a.

Cynthia Rudin, David Waltz, Roger N. Anderson, Albert Boulanger, Ansaf Salleb-Aouissi, Maggie Chow, Haimonti Dutta, Philip Gross, Bert Huang, Steve Ierome, Delfina Isaac, Arthur Kressner, Rebecca J. Passonneau, Axinia Radeva, and Leon Wu. Machine learning for the New York City power grid. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(2):328–345, Feb 2012b.

Cynthia Rudin, Şeyda Ertekin, Rebecca Passonneau, Axinia Radeva, Ashish Tomar, Boyi Xie, Stanley Lewis, Mark Riddle, Debbie Pangsrivinij, and Tyler McCormick. Analytics for Power Grid Distribution Reliability in New York City. accepted, 2014.

Lorenza Saitta and Filippo Neri. Learning in the "real world". *Machine Learning*, 30: 133–163, 1998.

Vignesh Veppur Sankaranarayanan, Junaed Sattar, and Laks VS Lakshmanan. Auto-play: A data mining approach to odi cricket simulation and prediction. In *Proceedings of the 2014 SIAM International conference on data mining*, pages 1064–1072. SIAM, 2014.

Robert E. Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, pages 1651–1686, 1998.

RobertP. Schumaker, OsamaK. Solieman, and Hsinchun Chen. Predictive modeling for sports and gaming. In *Sports Data Mining*, volume 26 of *Integrated Series in Information Systems*, pages 55–63. Springer US, 2010. ISBN 978-1-4419-6729-9.

Noam Shental, Aharon Bar-Hillel, Tomer Hertz, and Daphna Weinshall. Computing Gaussian mixture models with EM using equivalence constraints. In *Proceedings of Neural Information Processing Systems*, volume 16, pages 465–472, 2004.

Pannagadatta K Shivaswamy, Chiranjib Bhattacharyya, and Alexander J Smola. Second order cone programming approaches for handling missing and uncertain data. *The Journal of Machine Learning Research*, 7:1283–1314, 2006.

Edward H Simpson. The interpretation of interaction in contingency tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 238–241, 1951.

Aarti Singh, Robert Nowak, and Xiaojin Zhu. Unlabeled data: Now it helps, now it doesn't. In *Proceedings of Neural Information Processing Systems*, pages 1513–1520, 2008.

Maurice Sion. On general minimax theorems. *Pacific Journal of Mathematics*, 8(1): 171–176, 1958.

Brian Skinner. The problem of shot selection in basketball. *PloS one*, 7(1):e30776, 2012.

Mihailo Stojnic. Various thresholds for l1-optimization in compressed sensing. *arXiv preprint arXiv:0907.3666*, 2009.

Leanne Streja. *Models for Motorcycle Grand Prix Racing.* PhD thesis, University of California, Los Angeles, 2012.

Ichiro Takeuchi, Quoc V Le, Timothy D Sears, and Alexander J Smola. Nonparametric quantile estimation. *The Journal of Machine Learning Research*, 7:1231–1264, 2006.

Michel Talagrand. *The Generic Chaining*. Springer, 2005.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996. ISSN 0035-9246.

Geofrey G Towell, Jude W Shavlik, and M Noordewier. Refinement of approximate domain theories by knowledge-based neural networks. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, pages 861–866. Boston, MA, 1990.

B. S. Tsirelson, I. A. Ibragimov, and V. N. Sudakov. Norms of gaussian sample functions. In *Proceedings of the Third Japan–U.S.S.R. Symposium on Probability Theory. Lecture Notes in Math.*, volume 550, pages 20–41. Springer, 1976.

Theja Tulabandhula and Cynthia Rudin. Machine learning with operational costs. In *Proceedings of the International Symposium on Artificial Intelligence and Mathematics*, 2012.

Theja Tulabandhula and Cynthia Rudin. Machine learning with operational costs. *Journal of Machine Learning Research*, 14:1989–2028, 2013. URL `http://jmlr.org/papers/v14/tulabandhula13a.html`.

Theja Tulabandhula and Cynthia Rudin. On combining machine learning with decision making. *Machine Learning*, 2014.

Theja Tulabandhula, Cynthia Rudin, and Patrick Jaillet. The machine learning and traveling repairman problem. In Ronen I. Brafman, Fred S. Roberts, and Alexis Tsoukiàs, editors, *ADT*, volume 6992 of *Lecture Notes in Computer Science*, pages 262–276. Springer, 2011.

Office of Electric Transmission United States Department of Energy and Distribution. Grid 2030: A national vision for electricity's second 100 years. Technical report, United States, July 2003.

Ian Urbina. Mandatory safety rules are proposed for electric utilities. *New York Times*, 2004. August 21, Late Edition, Section B, Column 3, Metropolitan Desk, Page 2.

Robert J. Vanderbei. *Linear Programming: Foundations and Extensions, Third Edition*. Springer, 2008.

Vladimir N Vapnik. *Statistical learning theory*. Wiley, 1998.

Santosh Vempala. Geometric random walks: a survey. *MSRI Volume on Combinatorial and Computational Geometry*, 52:577–616, 2005.

Martin Wainwright. *Metric entropy and its uses (Chapter 3)*. Unpublished draft, 2011.

Andrés Weintraub, J. Aboud, C. Fernandez, G. Laporte, and E. Ramirez. An emergency vehicle dispatching system for an electric utility in Chile. *Journal of the Operational Research Society*, pages 690–696, 1999.

Huan Xu, Constantine Caramanis, and Shie Mannor. Robustness and regularization of support vector machines. *Journal of Machine Learning Research*, 10:1485–1510, December 2009.

Hua Yu, Jianxin Chen, Xue Xu, Yan Li, Huihui Zhao, Yupeng Fang, Xiuxiu Li, Wei Zhou, Wei Wang, and Yonghua Wang. A systematic prediction of multiple drug-target interactions from chemical, genomic, and pharmacological data. *PLoS ONE*, 5(7), 2012.

Tong Zhang. Covering number bounds of certain regularized linear function classes. *Journal of Machine Learning Research*, 2:527–550, 2002.

Dengyong Zhou, Jason Weston, Arthur Gretton, Olivier Bousquet, and Bernhard Schölkopf. Ranking on data manifolds. In *Advances in Neural Information Processing Systems 16*, pages 169–176. MIT Press, 2004.

Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

Xiaojin Zhu. Semi-supervised learning literature survey. Technical report, Computer Sciences TR 1530, University of Wisconsin – Madison, December 2007.