Journal of Machine Learning Research 14 (2013) 1989-2028 Submitted 12/11; Revised 8/12; Published 7/13

Machine Learning with Operational Costs

Theja Tulabandhula THEJA@MIT.EDU
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

Cynthia Rudin RUDIN@MIT.EDU
MIT Sloan School of Management and Operations Research Center
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

Editor: John-Shawe Taylor

Abstract

This work proposes a way to align statistical modeling with decision making. 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.

Keywords: statistical learning theory, optimization, covering numbers, decision theory

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 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. In Section 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 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 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 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. 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:

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

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 $\{\tilde{x}_i\}_i$ as representing machines in a factory waiting to be repaired, where the first feature $\tilde{x}_{i,1}$ is the age of the machine, the second feature $\tilde{x}_{i,2}$ is the condition at its last inspection, etc. The value $f^*(\tilde{x}_i)$ is the predicted probability of failure for $\tilde{x}_i$. Policy π^* is the order in which the machines $\{\tilde{x}_i\}_i$ are repaired, which is chosen based on how likely they are to fail, that is, $\{f^*(\tilde{x}_i)\}_i$, 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 $\{f^*(\tilde{x}_i)\}_i$'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 (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:

$$\text{Optimistic Bias: } f^\circ \in \operatorname*{argmin}_{f \in \mathcal{F}^{unc}} \left[\sum_{i=1}^{n} l\left(f(x_i), y_i\right) + C_2 R(f) + C_1 \min_{\pi \in \Pi} \text{OpCost}\left(\pi, f, \{\tilde{x}_i\}_i\right) \right], \tag{2}$$

$$\text{Pessimistic Bias: } f^\circ \in \operatorname*{argmin}_{f \in \mathcal{F}^{unc}} \left[\sum_{i=1}^{n} l\left(f(x_i), y_i\right) + C_2 R(f) - C_1 \min_{\pi \in \Pi} \text{OpCost}\left(\pi, f, \{\tilde{x}_i\}_i\right) \right]. \tag{3}$$

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 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 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 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).

2.1 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 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 data sets, 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 1(b) illustrates this possibility, by showing that 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.

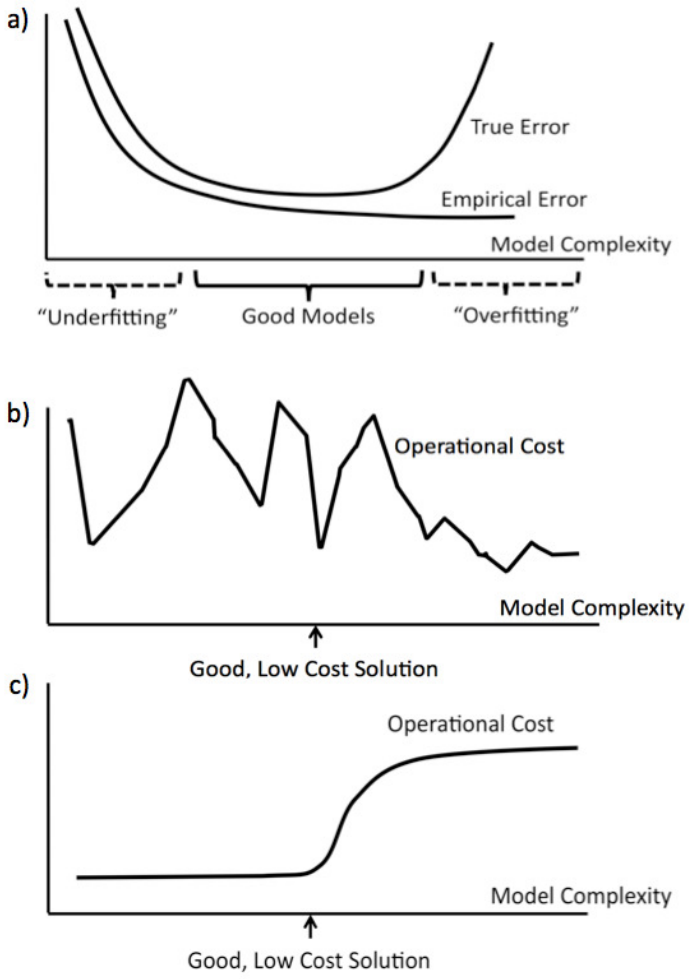

Figure 1: In all three plots, the x-axis represents model classes with increasing complexity. a) Relationship between training error and test error as a function of model complexity. b) A possible operational cost as a function of model complexity. c) Another possible operational cost.

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 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 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 5 we develop results of this type. These results indicate that in some cases, the operational cost can be an important quantity for generalization.

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 (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 paper.

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. 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, 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 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

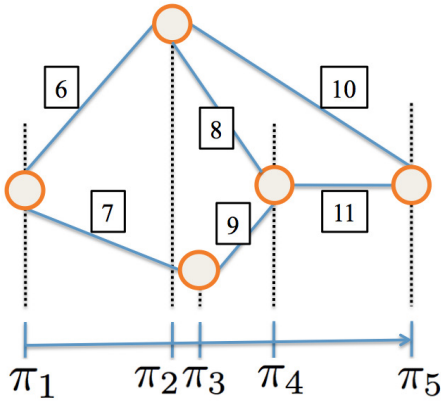

Figure 2: Staffing estimation with bias on scheduling with precedence constraints.

$(\beta^T x_i)$ 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:

$$\min_{\{\beta: \|\beta\|_2^2 \le C_2^*\}} \sum_{i=1}^{n} (y_i - \beta^T x_i)^2 + C_1 \min_{\pi \in \Pi(\beta, \{\tilde{x}_i\}_i)} (\pi_5 - \pi_1), \tag{4}$$

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

Figure 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 $\sim$16%. The r^2 values for both training and test vary much less, by $\sim$3.5%, where the best value happened to have the largest value of C_1. For small data sets, 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 $\sim$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.

1. Gurobi is the Gurobi Optimizer v3.0 from 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.

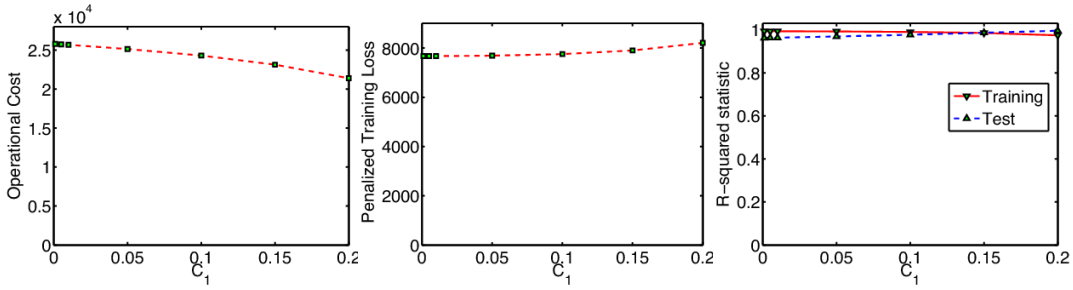

Figure 3: *Left:* Operational cost vs C_1. *Center:* Penalized training loss vs C_1. *Right:* R-squared statistic. $C_1 = 0$ corresponds to the baseline, which is the sequential formulation.

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 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, and thus at least the average of them:

$$\begin{aligned}\alpha \geq & \min_{\pi \in \Pi\{\beta, \{\tilde{x}_i\}_i\}} \pi_5 - \pi_1 \\ & \geq \max\{(\tilde{x}_6 + \tilde{x}_{10})^T\beta, (\tilde{x}_6 + \tilde{x}_8 + \tilde{x}_{11})^T\beta, (\tilde{x}_7 + \tilde{x}_9 + \tilde{x}_{11})^T\beta\} \\ & \geq z^T\beta \end{aligned} \tag{5}$$

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 5, Theorem 6, is a learning theoretic guarantee in the presence of this kind of arbitrary linear constraint, $z^T\beta \leq \alpha$.

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:

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

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 (6) in a cleaner way that was not possible directly with (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

$$\beta\in\{\beta:\beta\in\mathbb{R}^{13},\|\beta\|_2^2\le C_2^*\}$$

$$\pi\in\left\{\pi:\pi\in\{0,1\}^6,\sum_{i=1}^{6}\pi_i\le 3\right\}. \tag{7}$$

To solve (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 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 $\sim$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.

Before moving to the next application of the proposed framework, we provide a bound analogous to that of (5). Let us replace the soft constraint represented by the second term of (6) with a hard constraint and then obtain a lower bound:

$$\alpha\ge\max_{\pi\in\{0,1\}^6,\sum_{i=1}^{6}\pi_i\le 3}\sum_{i=1}^{6}(\beta^T\tilde{x}_i)\pi_i\ge\sum_{i=1}^{6}(\beta^T\tilde{x}_i)\pi_i', \tag{8}$$

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:

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

3. The version of the toolbox used is Version 5.1, Matlab R2010b, Mathworks, Inc.

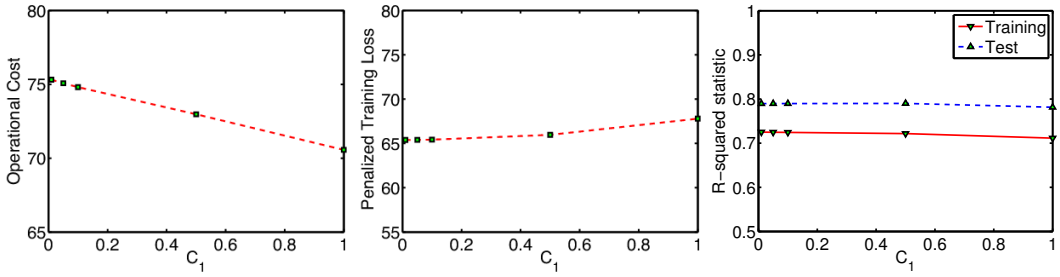

Figure 4: *Left:* Operational cost (total value) vs C_1. *Center:* Penalized training loss vs C_1. *Right:* R-squared statistic. $C_1 = 0$ corresponds to the baseline, which is the sequential formulation.

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

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 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.

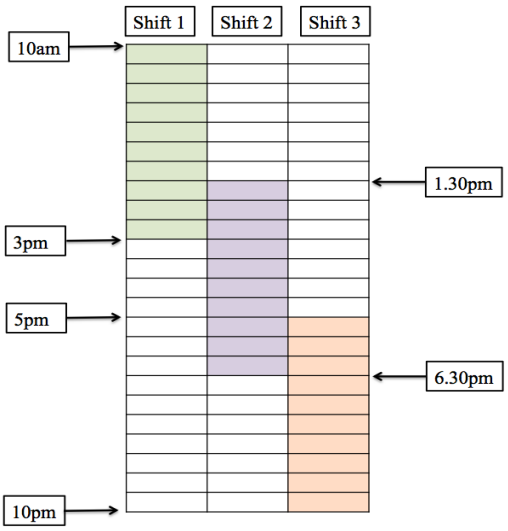

Figure 5: The three shifts for the call center. The cells represent half-hour periods, and there are 24 periods per work day. Work starts at 10am and ends at 10pm.

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 data set 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:

$$\min_{\beta:\|\beta\|_2^2 \le C_2^*} \sum_{i=1}^{n} (y_i - \beta^T x_i)^2 + C_1 \left[\min_{\pi} \sum_{i=1}^{3} \pi_i \text{ subject to } a_i^T \pi \ge (\beta^T \tilde{x}_i)^2 \text{ for } i = 1,\ldots,24, \pi \in \mathbb{Z}_+^3 \right], \tag{9}$$

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

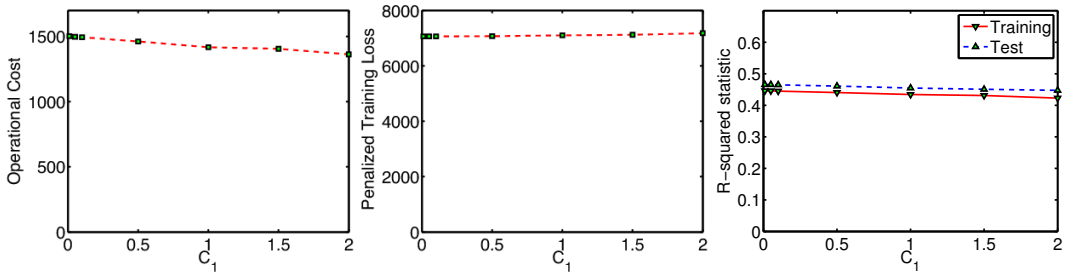

Figure 6: *Left:* Operational cost vs C_1. *Center:* Penalized training loss vs C_1. *Right:* R-squared statistic. $C_1 = 0$ corresponds to the baseline, which is the sequential formulation.

implementation of the Nelder-Mead algorithm, where at each step, Gurobi was used to solve the mixed-integer subproblem for finding the policy.

Figure 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 $\sim$1.6% and $\sim$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 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 (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:

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

Here α is related to the choice of C_1 and is fixed. $(\dagger)$ 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. $(\ddagger)$ 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 (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.

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, 2012, 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 5 (and in order to address question Q3 of Section 1).

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

4. For more on Robust Optimation see http://en.wikipedia.org/wiki/Robust_optimization.

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 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.

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, 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 (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 pre-determined, known and fixed. Let the optimization problem for the policy decision π be defined by:

$$\min_{\pi \in \Pi(f;\{\tilde{x}\}_i)} \text{OpCost}(\pi, f; \{\tilde{x}_i\}), \qquad \textit{(Base problem)} \tag{10}$$

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, since we have allowed the constraint set Π to be a function of both f and $\{\tilde{x}_i\}_i$, whereas in (2) and (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 (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, 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 (10), we can use an auxiliary variable, say t, to obtain an equivalent problem:

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

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 (10) can then be written as:

$$\min_{\pi \in \bigcap_{g \in \mathcal{F}_{good}} \Pi(g;\{\tilde{x}\}_i)} \max_{f \in \mathcal{F}_{good}} \text{OpCost}(\pi, f; \{\tilde{x}_i\}) \qquad \textit{(Robust counterpart)} \tag{11}$$

where we obtain a "robustly feasible solution" that is guaranteed to remain feasible for all values of $f \in \mathcal{F}_{good}$. In general, (11) is much harder to solve than (10) and is a topic of much interest in the robust optimization community. As we discussed earlier, there is no focus in (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 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:

$$\mathcal{F}_{good} = \left\{ f : f \in \mathcal{F}^R, \sum_{i=1}^{n} l(f(x_i), y_i) \leq \sum_{i=1}^{n} l(f^*(x_i), y_i) + \varepsilon \right\}, \tag{12}$$

for some fixed positive real ε. In (12), again f^* is a solution that minimizes the objective in (1) over $\mathcal{F}^{unc}$. The right hand side of the inequality in (12) is thus constant, and we will henceforth denote it with a single quantity C_1^*. Substituting this definition of $\mathcal{F}_{good}$ in (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:

$$\min_{\pi\in\Pi} \max_{\{f\in\mathcal{F}^R:\sum_{i=1}^n l(f(x_i),y_i)\le C_1^*\}} \left[\text{OpCost}\left(\pi, f, \{\tilde{x}_i\}_i\right)\right] \quad \textit{(Robust counterpart with assumptions)} \quad (13)$$

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) and (3):

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

$$\min_{f\in\mathcal{F}^{unc}} \left[\sum_{i=1}^n l\left(f(x_i),y_i\right) + C_2 R(f) - C_1 \min_{\pi\in\Pi} \text{OpCost}\left(\pi, f, \{\tilde{x}_i\}_i\right)\right] \textit{(Simultaneous pessimistic)}. \quad (14)$$

Apart from the assumption **A1** on the decision set Π that we made in (13), we will also assume that $\mathcal{F}_{good}$ defined in (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 (14) and (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 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)* OpCost$(\pi, f, \{\tilde{x}_i\}_i)$ *be concave continuous in f and convex continuous in π (assumption **A3**). Then, the robust optimization problem (13) is equivalent to the pessimistic bias optimization problem (14). That is,*

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

Remark 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 1, we restate a well-known generalization of von Neumann's minimax theorem and some related definitions.

Definition 3 *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 $\varepsilon > 0$ there exists a neighborhood U of p_0 such that $h(p) \leq h(p_0) + \varepsilon$ 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 4 *(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 (1).

Proof *(Of Proposition 1)* We start from the left hand side of the equivalence we want to prove:

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

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 4) 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 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.

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 (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 (1). Then the definition of $\mathcal{F}_{good}$ in terms of the least squares loss is:

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

Since each $f \in \mathcal{F}_{good}$ corresponds to at least one β, the optimization of (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 (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 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 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 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.

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

Table 1: Table showing a summary of different possible uncertainty set descriptions that are based on ML loss functions.

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 1 do not hold for robust optimization problems that use these sets.

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 (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 (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 1 that pessimistic bias on operational 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 7 represents the overlap between the three classes of

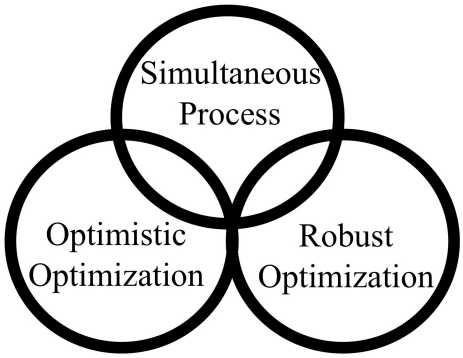

Figure 7: Set based description of the proposed framework (top circle) and its relation to robust (right circle) and optimistic (left circle) optimizations. The regions of intersection are where the conditions on the objective OpCost and the feasible set Π are satisfied.

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 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 3 lies within the intersection of optimistic optimization and the simultaneous process; this can be deduced from (7).

In Section 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.

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?"

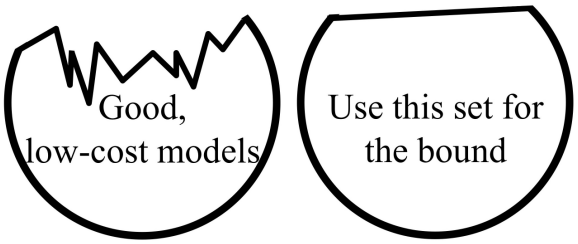

Figure 8: Left: hypothesis space for intersection of good models (circular, to represent ℓ_q ball) with low cost models (models below cost threshold, one side of wiggly curve). Right: relaxation to intersection of a half space with an ℓ_q ball.

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 10.

In Section 3, we showed that a bias on the operational cost can sometimes be transformed into linear constraints on model parameter β (see Equations (5) and (8)). There is a broad class of other problems for which this is true, for example, for applications related to those presented in Section 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 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 6). Following that, we will build up to a generalization bound (Theorem 10) that incorporates Theorem 6. After that will be the proof of Theorem 6.

Definition 5 *(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 $\varepsilon > 0$, 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 $d_\rho(a,u) \leq \varepsilon$.*
- *The **covering number** of A is $N(\varepsilon, A, \rho) := \inf_U |U|$ 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 $X = [x_1 \ \ldots \ x_n]^T$ with $h_j \in \mathbb{R}^n$, $j = 1, \ldots, p$,

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 $\{c_{j\nu}\}_{j,\nu}$:

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

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

$$P^K := \left\{ (k_1, ..., k_p) \in \mathbb{Z}^p : \sum_{j=1}^{p} |k_j| \leq K \right\}.$$
$$P_c^K := \left\{ (k_1, ..., k_p) \in P^K : \sum_{j=1}^{p} \tilde{c}_{j\nu} k_j \leq K \; \forall \nu = 1, ..., V \right\}. \tag{15}$$

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 $\tilde{X}$ 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 $\lambda_{\min}(\tilde{X}^T \tilde{X})$ to be the smallest eigenvalue of the matrix $\tilde{X}^T \tilde{X}$, which will thus be non-negative. Using these definitions, we state our main result of this section.

Theorem 6 *(Main result, covering number bound)*

$$N(\sqrt{n}\varepsilon, \mathcal{F}_{|S}, \|\cdot\|_2) \leq \begin{cases} \min\{|P^{K_0}|, |P_c^K|\} & \text{if } \varepsilon < X_b B_b \\ 1 & \text{otherwise} \end{cases}, \tag{16}$$

where

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

and

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

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_{jv} on the operational cost become tighter. In other words, as the c_{jv} 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 10.

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 $\sqrt{n}\varepsilon$ to cover $\mathcal{F}_{|S}$, so $N(\sqrt{n}\varepsilon, \mathcal{F}_{|S}, \|\cdot\|_2) = 1$. In that case, the covering number in Theorem 6 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, $\lambda_{\min}(\tilde{X}^T\tilde{X})$ 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 $\lambda_{\min}(\tilde{X}^T\tilde{X}) = 0$, 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. 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

$$R^{\text{true}}(l \circ f) \le R^{\text{emp}}(l \circ f, \{(x_i,y_i)\}_1^n) + \text{Bound}(\text{complexity}(\mathcal{F}),n), \tag{17}$$

where the complexity term takes into account the constraints on $\mathcal{F}$, both the linear constraints, and the ℓ_q-ball constraint. Theorem 6 gives an upper bound on the term Bound(complexity($\mathcal{F}$),n) in (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 7 *(Rademacher Complexity) The empirical Rademacher complexity of $\mathcal{F}_{|S}$ is*[5]

$$\hat{\mathcal{R}}(\mathcal{F}_{|S}) = \mathbb{E}_\sigma\left[\sup_{f\in\mathcal{F}}\frac{2}{n}\sum_{i=1}^n \sigma_i f(x_i)\right] \tag{18}$$

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

5. The factor 2 in the defining equation (18) is not very important. Some authors omit this factor and include it explicitly as a pre-factor in, for example, Theorem 8.

The empirical Rademacher complexity $\hat{\mathcal{R}}(\mathcal{F}_{|S})$ can be computed given S and $\mathcal{F}$, 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 8 *(Rademacher Generalization Bound) For all $\delta > 0$, with probability at least $1-\delta, \forall h \in \mathcal{H}$,*

$$R^{\text{true}}(l \circ h) \leq R^{\text{emp}}(l \circ h, \{x_i, y_i\}_1^n) + \mathcal{L} \cdot \hat{\mathcal{R}}(\mathcal{H}_{|S}) + \frac{3}{\sqrt{2}} \sqrt{\frac{\log \frac{1}{\delta}}{n}}, \tag{19}$$

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

Note that (19) is an explicit form of (17). We will now relate $\hat{\mathcal{R}}(\mathcal{F}_{|S})$ to covering numbers thus justifying the importance of statement (16) in Theorem 6. 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 9 *(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,*

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

Our main result in Theorem 6 can be used in conjunction with Theorems 8 and 9, 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 10 *(Generalization Bound for ML with Operational Costs) For all $\delta > 0$, with probability at least $1-\delta, \forall f \in \mathcal{F}$,*

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

where

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

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

and

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

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_{jv}. 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 $\varepsilon = 0$ to $\varepsilon = \infty$. When ε is larger than $X_b B_b$, as noted earlier, $N(\sqrt{n}\varepsilon, \mathcal{F}_{|S}, \|\cdot\|_2) = 1$ and thus $\log N(\sqrt{n}\varepsilon, \mathcal{F}_{|S}, \|\cdot\|_2) = 0$.

Before we move onto building the necessary tools to prove Theorem 6, 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 6 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.

5.1 Proof of Theorem 6

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 6 adapts Maurey's Lemma to handle polyhedrons, and allows us to apply counting techniques to bound the covering number.

Recall that $X = [x_1 \ldots x_n]^T$ was also defined column-wise as $[h_1 \ \ldots \ h_p]$. 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:

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

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 11 *Let* $\max_{j=1,\ldots,p} \|\tilde{h}_j\|$ *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 $j = 1, \ldots, p$, $\bar{\gamma}_j \geq 0$ 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 $\{\tilde{h}_1, \ldots, \tilde{h}_p, \mathbf{0}\}$. That is, $\Pr(\tilde{\mathsf{h}} = \tilde{h}_j) = \bar{\gamma}_j$, $j = 1, \ldots, p$ and $\Pr(\tilde{\mathsf{h}} = \mathbf{0}) = \bar{\gamma}_{p+1}$.

Draw K observations $\{\tilde{h}^1, \ldots, \tilde{h}^K\}$ 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:

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}}[\|\mathsf{y}_K - y\|^2] &= \mathbb{E}_{\mathcal{D}}[\|\mathsf{y}_K - \mathbb{E}_{\mathcal{D}}[\mathsf{y}_K]\|^2] = \mathbb{E}\left[\sum_{i=1}^{n}(\mathsf{y}_K - \mathbb{E}_{\mathcal{D}}[\mathsf{y}_K])_i^2\right] \\
&\overset{(\dagger)}{=} \sum_{i=1}^{n}\mathrm{Var}((\mathsf{y}_K)_i) \overset{(*)}{=} \sum_{i=1}^{n}\frac{1}{K}\mathrm{Var}((\tilde{\mathsf{h}})_i) \\
&\overset{(\ddagger)}{=} \frac{1}{K}\sum_{i=1}^{n}\left(\mathbb{E}_{\mathcal{D}}[(\tilde{\mathsf{h}})_i^2] - \mathbb{E}_{\mathcal{D}}[(\tilde{\mathsf{h}})_i]^2\right) \overset{(\circ)}{=} \frac{1}{K}\left(\mathbb{E}_{\mathcal{D}}[\|\tilde{\mathsf{h}}\|^2] - \|\mathbb{E}_{\mathcal{D}}[\tilde{\mathsf{h}}]\|^2\right) \\
&\leq \frac{1}{K}\mathbb{E}_{\mathcal{D}}[\|\tilde{\mathsf{h}}\|^2] \leq \frac{b^2}{K} \qquad (20)
\end{aligned}$$

where we have used i to be the index for the i^{th} coordinate of the n dimensional vectors. $(\dagger)$ follows from the definition of variance coordinate-wise. $(*)$ follows because each component of y_K is a sample average. $(\ddagger)$ 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 $\max_{j=1,\ldots,p}\|\tilde{h}_j\|$ tells us that $\mathbb{E}_{\mathcal{D}}[\|\tilde{\mathsf{h}}\|^2] \leq b^2$ leading to (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 12 *For any y and K as considered above, we can find non-negative integers $m_1, \ldots, m_p$ 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 11, 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 $m_1, \ldots, m_p$ gives us a covering of the set that y belongs to. In the case of Lemma 11, 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 13 *Let $\max_{j=1,\ldots,p}\|\tilde{h}_j\|$ 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 $k_1, \ldots, k_p$ such that $\sum_{j=1}^{p}|k_j| \leq K$.

Proof Lemma 11 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 11 and Corollary 12 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 $\tilde{X}$, defined before Theorem 6. It has the elements of the scaled set $\{\tilde{h}_j\}_j$ as its columns: $\tilde{X} := [\tilde{h}_1 \ \ldots \ \tilde{h}_p]$.

Lemma 14 *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:*

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

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

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

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 $\kappa := [k_1/K \ \dots \ k_p/K]^T$. Using the definition of $\tilde{X}$,

$$\begin{aligned} \frac{b^2}{K} &\geq \|y - y_K\|_2^2 = \|\tilde{X}\tilde{\beta} - \tilde{X}\kappa\|_2^2 = \|\tilde{X}(\tilde{\beta} - \kappa)\|_2^2 \\ &= (\tilde{\beta} - \kappa)^T \tilde{X}^T \tilde{X} (\tilde{\beta} - \kappa) \overset{(*)}{\geq} \lambda_{\min}(\tilde{X}^T \tilde{X}) \|\tilde{\beta} - \kappa\|_2^2. \end{aligned} \tag{21}$$

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, $|z_j| \leq \max_{j=1,...,p} |z_j| = \|z\|_\infty \leq \|z\|_2$. Combining this with (21), we have:

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

This implies that κ itself component-wise satisfies

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

So far we know that for all $\nu = 1, ..., V$, $\sum_{j=1}^{p} \tilde{c}_{j\nu} \tilde{\beta}_j + \delta_\nu \leq 1$, 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 $\sum_{j=1}^{p} \tilde{c}_{j\nu} \frac{k_j}{K} \leq 1, \ \nu = 1, ..., V$; 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 $\nu = 1, ..., V$. That is,

$$\begin{aligned} &\frac{b}{\sqrt{K \lambda_{\min}(\tilde{X}^T \tilde{X})}} = A \leq \min_{\nu=1,...,V} \frac{\delta_\nu}{\sum_{j=1}^{p} |\tilde{c}_{j\nu}|} \\ \Leftrightarrow \ \ &K \geq \frac{b^2}{\left[\min_{\nu=1,...,V} \frac{\delta_\nu}{\sum_{j=1}^{p} |\tilde{c}_{j\nu}|} \right]^2 \lambda_{\min}(\tilde{X}^T \tilde{X})}. \end{aligned}$$

■

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 6)*

Recall that

- the matrix X is defined as $[h_1 \ ... \ h_p]$;
- 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 $j = 1, ..., p$;
- 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 $j = 1, ..., p$; 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:

$$\begin{aligned} \sum_{j=1}^p |\tilde{\beta}_j| =& \frac{1}{n^{1/r} B_b X_b} \sum_{j=1}^p \|h_j\|_r |\beta_j| \\ \leq& \frac{1}{n^{1/r} B_b X_b} \left(\sum_{j=1}^p \|h_j\|_r^r \right)^{1/r} \left(\sum_{j=1}^p |\beta^j|^q \right)^{1/q}. \end{aligned} \tag{22}$$

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, that is, $(h_j)_i$ 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 (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 6. Facts 1 and 2 show that the requirements on $\tilde{\beta}$ for Lemma 13 and Lemma 14 are satisfied. Fact 3 shows that the requirement on $\{\tilde{h}_j\}_j$ for Lemma 13 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 14 is satisfied and also we would like the squared distance between y and y_K to be less than $n\varepsilon^2$. To do this, we pick K to be the bigger of the two quantities: $X_b^2 B_b^2/\varepsilon^2$ and that given in Lemma 14. That is,

$$K = \left\lceil \max\left\{ \frac{X_b^2 B_b^2}{\varepsilon^2}, \frac{nX_b^2 B_b^2}{\left[\min_{\nu=1,...,V}\frac{\delta_\nu}{\sum_{j=1}^{p}|\tilde{c}_{j\nu}|}\right]^2 \lambda_{\min}(\tilde{X}^T\tilde{X})} \right\} \right\rceil. \tag{23}$$

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 $y_K = \sum_{j=1}^{p} \frac{k_j}{K} \tilde{h}_j$ is $\varepsilon\sqrt{n}$ close to $y = \sum_{j=1}^{p} \tilde{\beta}_j \tilde{h}_j$. That is,

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

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 13 where we have also used K satisfying $K \geq X_b^2 B_b^2/\varepsilon^2 \geq 1$. Lemma 14 along with the choice of K from (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 $\varepsilon\sqrt{n}$ 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 (15), gives a bound on the covering number, which is $|P_c^K|$. That is,

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

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

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

where $K_0 := \left\lceil \frac{X_b^2 B_b^2}{\varepsilon^2} \right\rceil$ by using Lemma 13 and very similar arguments as above.

In addition, when $\varepsilon \geq X_b B_b$, 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:

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

■

Since Theorem 6 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.

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 3), "Can we produce a reasonable probabilistic model, supported by data, where we might expect to pay a specific amount?" (answered in Section 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 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 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 paper 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 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 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 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.

Acknowledgments

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

References

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.

Martin Anthony and Peter L. Bartlett. *Neural network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

Kevin Bache and Moshe Lichman. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

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.

Russell R. Barton, Barry L. Nelson, and Wei Xie. A framework for input uncertainty analysis. In *Winter Simulation Conference*, pages 1189–1198. WSC, 2010.

John R. Birge and François Louveaux. *Introduction to Stochastic Programming*. Springer Verlag, 1997.

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.

Lawrence D. Brown, Ren Zhang, and Linda Zhao. Root-unroot methods for nonparametric density estimation and poisson random-effects models. *Department of Statistics University of Pennsylvania, Tech. Rep*, 2001.

Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

Jeśus A. De Loera. The many aspects of counting lattice points in polytopes. *Mathematische Semesterberichte*, 52(2):175–195, 2005.

Simon French. *Decision Theory: An Introduction to the Mathematics of Rationality*. Halsted Press, 1986.

Sven Ove Hansson. *Decision Theory: A Brief Introduction*. Online manuscript. Department of Philosophy and the History of Technology, Royal Institute of Technology, Stockholm, 1994.

Yaochu Jin. *Multi-Objective Machine Learning, In Studies in Computational Intelligence*, volume 16. Springer, 2006.

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.

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.

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.

David Pollard. *Convergence of Stochastic Processes*. Springer, 1984.

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, 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 2012.

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.

Maurice Sion. On general minimax theorems. *Pacific Journal of Mathematics*, 8(1):171–176, 1958.

Michel Talagrand. *The Generic Chaining*. Springer, 2005.

Theja Tulabandhula and Cynthia Rudin. Machine learning with operational costs. In *Proceedings of the International Symposium on Artificial Intelligence and Mathematics*, 2012.

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.

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 Naumovich Vapnik. *Statistical Learning Theory*, volume 2. Wiley New York, 1998.

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.

Xiaojin Zhu. Semi-supervised learning literature survey. Technical report, Computer Sciences TR 1530, University of Wisconsin Madison, December 2007.