The Influence of Operational Cost on Estimation

Theja Tulabandhula and Cynthia Rudin
Massachusetts Institute of Technology
Cambridge MA 02139
{theja,rudin}@mit.edu

Abstract

This work concerns the way that statistical models are used to make decisions. In particular, we aim to merge the way estimation algorithms are designed with how they are used for a subsequent task. Our methodology considers the operational cost of carrying out a policy, based on a predictive model. The operational cost becomes a regularization term in the learning algorithm's objective function, allowing either an *optimistic* or *pessimistic* view of possible costs. Limiting the operational cost reduces the hypothesis space for the predictive model, and can thus improve generalization. We show that different types of operational problems can lead to the same type of restriction on the hypothesis space, namely the restriction to an intersection of an ℓ_q ball with a halfspace. We bound the complexity of such hypothesis spaces by proposing a technique that involves counting integer points in polyhedrons.

Keywords: statistical learning theory, optimization, covering numbers.

1 Introduction

Decisions are usually made using a *sequential* process: first, data are input into a statistical algorithm to produce a predictive model, and that model is used to make recommendations for the future; second, the user develops a plan of action, based on how feasible it is to carry out the recommendations. For example, in scheduling staff for a medical clinic, recommendations based on a statistical model of the number of patients might be used to determine the policy for staffing. In traffic flow problems, recommendations based on a model of the forecasted traffic might afterwards be used to determine load balancing policies on the network. In online advertising, recommendations based on models for the payoff and ad-click rate might be used to determine a policy for when the ad should be displayed. When deciding how to follow the algorithm's recommendations, some actions may be easier to implement than others, and perhaps it is more cost effective to follow certain recommendations but not others. Current algorithms only produce a statistical prediction; they do not know whether the recommendations they make will lead to feasible or actionable policies. The users simply do the best they can to follow the recommendations of the statistical model, generally without questioning whether the model itself could be altered in order to make more practical recommendations.

The goal of this work is to merge the way machine learning algorithms are designed with how their estimations are used for a subsequent task. We would like to have statistical algorithms that possess an inherent business intelligence, in that they can "understand" an organization's various operational costs in order to make better recommendations. We use a *simultaneous process* rather than a *sequential process*, where the predictive model and the policy are determined at the same time. Our algorithms directly consider uncertainty in the predictive process (which arises because we have a finite sample). For example, these algorithms could choose, among all approximately equally good predictive models, the one that minimizes the organization's cost of carrying out the policy based on the model's recommendations. This would be an optimistic view, in that it would allow the organization to carry out a cost-effective policy while still being true to the historical data. Business managers often like to know if there is some scenario that is supported by the data, but for which the operational cost is low; our algorithms would be able to find such scenarios. In that sense, our algorithms encode "erring on the side of caution" in terms of overspending. For other types of problems, we can design algorithms to choose, among all reasonable models, one that is pessimistic (even robust) in terms of operational costs, which would help to ensure that the organization does not underestimate the operational cost and under-allocate. The key point is that the algorithms are not oblivious to the operational costs, as is done traditionally. For many problems, there is often no one right statistical model (unless the statistical problem is very easy), and our framework allows us to take this uncertainty into account in order to choose a more *practical* model.

This idea can provide a substantial benefit in some cases; it is possible that a small change in the predictive model can induce a large change in the operational cost, without decreasing predictive quality. We will introduce the simultaneous process formally in Section 2 and discuss the circumstances when this is likely to happen. In the simultaneous process, the regularization term of the machine learning algorithm encodes the decision and its associated operational cost. This means the regularization term itself can be the optimal value of a complicated optimization problem.

In Section 3, we give two examples of algorithms that incorporate these operational costs. Our first example application is a staffing problem at a medical clinic, where the goal 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.

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.

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. The simultaneous process essentially introduces a bias towards low or high operational cost, where "bias" means (as usual) a preference for certain desirable properties (*e.g.,* another type of bias is model sparsity [1]). In Section 5, we show first that the hypothesis spaces for both 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 [2, 3] 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.

We start by formalizing the "simultaneous" process, where operational costs are incorporated into machine learning algorithms.

2 The Sequential and Simultaneous Processes

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

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

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 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 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:

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

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

Step 2: Compute the policy:

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

When $C_1 = 0$, the simultaneous process becomes the sequential process. The optimization problem in the simultaneous process is often computationally more difficult than in the sequential version, 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 [4], Nelder-Mead [5] and Alternating Minimization schemes [6].

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 with sparse or smooth regularization terms, the new operational cost bias can be interpreted as a prior belief about the model - in this case, a belief that the operating costs should be lower or higher on the current set of unlabeled instances $\{\tilde{x}_i\}_i$. In that sense, we introduce a type of regularization that may have a closer connection to reality than typical (e.g., ℓ_1 or ℓ_2 norm) regularizers. If one asks a manager at a company what prior belief they have about the estimation model, it is not likely they would give a answer in terms of coefficients for a linear model. Even managers who are not mathematicians or computer scientists might have some belief - they could perhaps believe that they are expecting to spend a certain amount to enact the policy. It is possible that this type of belief, which relies on direct experience, might be more practical, and more accurate, than the more abstract prior information that we are typically used to dealing with. Further, the simultaneous method can be used to assist in pre-allocating costs. If there is some uncertainty about how much it will cost to solve a problem, the simultaneous method can be used to find a range of possible costs, from optimistic to pessimistic, which will determine how much should be allocated to solve the problem.

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.

We would like to clarify some things to avoid possible misperceptions about the general idea. First, we are not claiming that "truth" is altered by what one needs to do with it; one can view the operational cost term purely as encoding a prior belief about the truth. The prior belief happens to be about the operational cost. Second, what we call "operational cost" is essentially a form of utility that is used to bias the picture of the world towards anticipated decisions. One should not view the operating cost as a utility function that needs to be estimated, as in reinforcement learning, where we do not know the cost. It is possible to extend our framework to estimate the utility, but currently, the cost is fixed and there is no separate utility: one knows precisely what the cost will be under each possible outcome. For instance, if we are estimating prices, and then the price is revealed, we know exactly what we will pay.

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 [7], 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.

3 Conceptual Demonstrations

Throughout this section, we will assume that we are working with linear functions f of the form $\beta^T x$ so that $\Pi(f, \{\tilde{x}_i\}_i)$ is equivalent to $\Pi(\beta, \{\tilde{x}_i\}_i)$. 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 1. 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 1, 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. (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 is available as *manpower* in the R-package *bestglm*, using "Hour," "Load" and "Stay" columns. The training error is chosen to be the least squares loss between the estimated time for stations to finish their jobs ($\beta^T x_i$) and the actual times it took to finish (y_i). The unlabeled data are the new load and current patients present at each station for a given period, given as $\tilde{x}_6, .., \tilde{x}_{11}$. Let π denote the 5-dimensional real vector with coordinates $\pi_1, ..., \pi_5$.

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

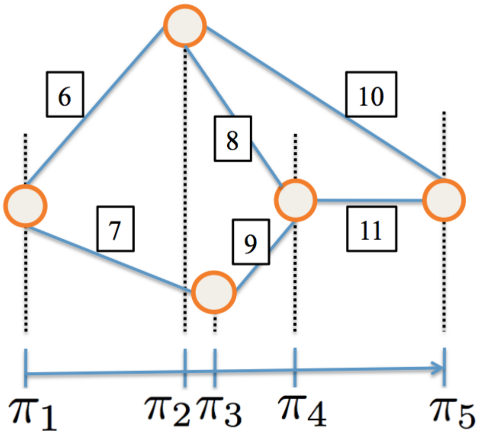

Figure 1: Staffing estimation with bias on scheduling with precedence constraints.

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

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

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

To solve (4) given values of C_1 and C_2, we used a function-evaluation-based scheme (Nelder-Mead [5]) 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 2 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 is reduced substantially, by ∼16%. The r^2 values for both training and test vary much less, by ∼3.5%, where the best value happened to have the largest value of C_1. For small datasets, there is generally a variation between training and test: for this data split, there is a 3.16% difference in r^2 between training and test for plain least squares, and this is similar across various splits of the training and test data. This means that for the scheduling problem, there is a range of reasonable predictive models within

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

[2] If $\hat{y}_i$ are the predicted labels and $\bar{y}$ is the mean of $\{y_1, ..., y_n\}$, then the value of the r^2 statistic is defined as $1 - \sum_i (y_i - \hat{y}_i)^2 / \sum_i (y_i - \bar{y})^2$.

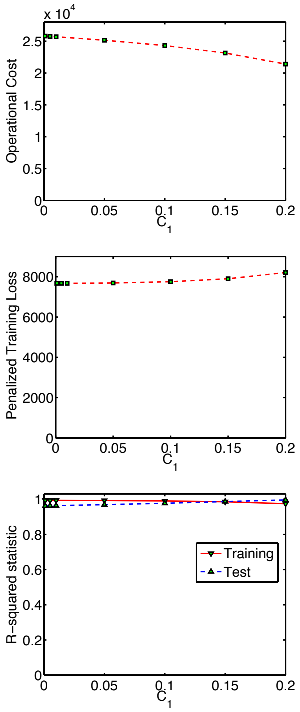

Figure 2: *Top:* Operational cost vs C_1. *Center:* Penalized training loss vs C_1. *Bottom:* R-squared statistic. $C_1 = 0$ corresponds to the baseline, which is the sequential formulation.

about ∼3.5% of each other. The optimistic bias allows a more cost-effective solution, staying within this regime of reasonable predictive models. If we are lucky, we can potentially save 16% of operational time.

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 3 paths in Figure 1, and thus at least the average of them:

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

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 5.1, 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. She will remodel them at fixed costs and sell them for a profit. The fixed costs for the 6 properties are denoted $\{c_i\}_{i=1}^6$. She estimates the value of each property from data regarding historical sales, in this case, from the *Boston Housing* data set [8], which has 13 features. Let policy $\pi \in \{0,1\}^6$ be the 6-dimensional binary vector that indicates the properties she purchases. 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, profit) is the sum of the three property values plus the costs for repair work. A pessimistic bias on profit is chosen to motivate a min-max formulation. The resulting (mixed-integer) program for Step 1 of the simultaneous process is:

$$\begin{aligned}&\min_{\beta\in\{\beta:\beta\in\mathbb{R}^{13},\|\beta\|_2^2\leq C_2^*\}} \sum_{i=1}^n (y_i-\beta^T x_i)^2 \\ &\quad + C_1\left[\max_{\pi\in\{0,1\}^6;\sum_{i=1}^6\pi_i\leq 3}\sum_{i=1}^6(\beta^T\tilde{x}_i+c_i)\pi_i\right] \end{aligned} \qquad (6)$$

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

$$\begin{aligned}&\beta \in \{\beta : \beta\in\mathbb{R}^{13}, \|\beta\|_2^2 \leq C_2^*\} \\ &\pi \in \left\{\pi : \pi\in\{0,1\}^6, \sum_{i=1}^6 \pi_i \leq 3\right\}. \end{aligned} \qquad (7)$$

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 3 shows the operational profit, the training loss, and r^2 values for a range of C_1. The training loss and r^2 values change by less than ∼3.5%, whereas the operational profit changes about 6.5%. The pessimistic bias shows that even if the developer chose the best response policy to the prices, she might make on the order of 6.5% less if she is unlucky.

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

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

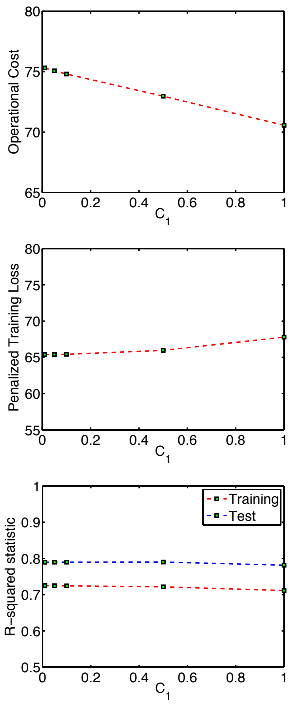

Figure 3: *Top:* Operational profit vs C_1. *Center:* Penalized training loss vs C_1. *Bottom:* R-squared statistic. $C_1 = 0$ corresponds to the baseline, which is the sequential formulation.

term 2 of (6) with a hard constraint and then obtain a lower bound:

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

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

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

This is again a linear constraint on the function class parametrized by β, that 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).

In the longer version of our work [9] we have considered two more applications: one, where the regularization term is the solution of a different type of scheduling problem; and second, the Machine Learning & Traveling Repairman Problem (ML&TRP) of [10]. In all cases, the simultaneous method performs about equally well for prediction as the sequential method, but gives a range of operational costs.

4 Connections to Robust Optimization

The goal of robust optimization (RO) is to provide the best possible policy that is acceptable under all possible conditions (*e.g.*, see [11]). For example, 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 [12] 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. 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 occur (which is what the pessimistic simultaneous formulation handles). Or, depending on the application, it may be better to choose a best response policy, to the outcome that is most likely to actually occur than the one that is aimed generally at all reasonable cases, including the worst case.

4.1 Equivalence between RO and the simultaneous method in some cases

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 method, using a minimax theorem.

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 method. 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:

$$\min_{\pi\in\Pi(f;\{\tilde{x}\}_i)} \text{OpCost}(\pi, f; \{\tilde{x}_i\}), \tag{9}$$

where $\Pi(f; \{\tilde{x}_i\})$ is the feasible set for the optimization problem. 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 (9) can then be written as:

$$\min_{\pi \in \bigcap\limits_{g\in\mathcal{F}_{good}} \Pi(g;\{\tilde{x}\}_i)} \max_{f\in\mathcal{F}_{good}} \text{OpCost}(\pi, f; \{\tilde{x}_i\}) \quad (10)$$

where we obtain a "robustly feasible solution" that is guaranteed to remain feasible for all values of $f \in \mathcal{F}_{good}$. In general, (10) is much harder to solve than (9) and is a topic of much interest in the robust optimization community (*e.g.,* see [11]). As we discussed earlier, there is no focus in (10) 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:

$$\mathcal{F}_{good} = \left\{ f : f \in \mathcal{F}^R, \sum_{i=1}^{n} l\left(f(x_i), y_i\right) \leq \sum_{i=1}^{n} l\left(f^*(x_i), y_i\right) + \epsilon \right\}, \quad (11)$$

for some fixed positive real ϵ. In (11), again f^* is a solution that minimizes the objective in (1) over $\mathcal{F}^{unc}$. The right hand side of the inequality in (11) is thus constant, and we will henceforth denote it with a single quantity C_1^*. Substituting this definition of $\mathcal{F}_{good}$ in (10), 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:

$$\min_{\pi\in\Pi} \max_{\{f\in\mathcal{F}^R:\sum_{i=1}^n l(f(x_i),y_i)\leq C_1^*\}} \left[\text{OpCost}\left(\pi, f, \{\tilde{x}_i\}_i\right)\right] \quad (12)$$

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

Apart from the assumption **A1** on the decision set Π that we made in (12), we will also assume that $\mathcal{F}_{good}$ defined in (11) 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 (3) and (12), 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 4.1 *Let $\Pi(f;\{\tilde{x}_i\}_i) = \Pi$ be compact, convex, and independent of parameters f and $\{\tilde{x}_i\}_i$ (assumption **A1**). Let $\{f \in \mathcal{F}^R : \sum_{i=1}^n l(f(x_i), y_i) \leq C_1^*\}$ 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 (12) is equivalent to the pessimistic bias optimization problem (3). That is,*

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

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

The equivalence relationship of Proposition 4.1 shows that there is a problem class in which each instance can be viewed either as a RO problem or an estimation problem with an operational cost bias. The proof is provided in the full version [9].

In Section 5, we will provide statistical guarantees for the simultaneous method. These are very different from the style of probabilistic guarantees in the robust optimization literature (*e.g.,* [13]). There are some "sample complexity" bounds [12] 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 method that involve integer point counting 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 is equivalent to imposing a particular bias; more bias equates to a better bound.

To establish the bound, it is sufficient to provide an upper bound on the covering number, since there are large number of probabilistic bounds in the learning theory literature (*e.g.*, [14]) that can be applied to obtain a generalization bound. 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

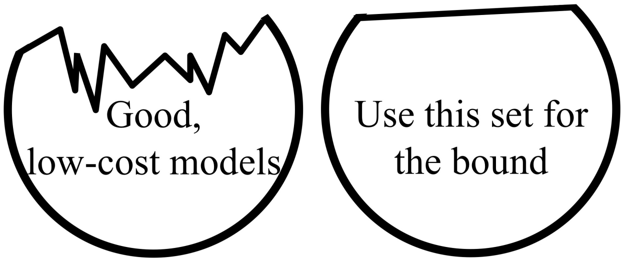

Figure 4: Left: hypothesis space for intersection of good models (circular, to represent ℓ_q ball) with low cost models (models below cost threshold, one side of wiggly curve). Right: relaxation to intersection of a half space with an ℓ_q ball.

of other problems for which this is true, for example, the ML&TRP application of [10] also has this property. 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 4.

Definition *(Covering Number, [15])* Let $A \subseteq X$ be an arbitrary set and (X, ρ) a (pseudo-)metric space. Let $|\cdot|$ denote set size.

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

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

$$\begin{aligned}
&\mathcal{F} := \{f : f(x) = \beta^T x, \beta \in \mathcal{B}\} \text{ where} \\
&\mathcal{B} := \{\beta \in \mathbb{R}^p : \|\beta\|_q \le B_b, \\
&\qquad \sum_{j=1}^p c_{jl}\beta_j + \delta_l \le 1, \delta_l > 0, l = 1, ..., L\},
\end{aligned}$$

where $1/r + 1/q = 1$ and $\{c_{jl}\}_{j,l}$, $\{\delta_l\}_l$ and B_b are known constants. Let $\mathcal{F}_{|S}$ be defined as the restriction of $\mathcal{F}$ with respect to S.

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

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

Define $\tilde{X}$ to be equal to X times a diagonal matrix whose j^{th} diagonal element is $\frac{n^{1/r} X_b B_b}{\|h_j\|_r}$. Let K be a positive number. Further, let the set P_c^K be defined as the set

$$\begin{aligned}
\Bigg\{ \{k_i\}_{i=1}^p \in \mathbb{Z}^p : \sum_{j=1}^p |k_j| \le K, && (13) \\
\sum_{j=1}^p \tilde{c}_{jl} k_j \le K \ \forall l = 1, \dots, L \Bigg\}.
\end{aligned}$$

Let count(P_c^K) be size of set P_c^K. Using these definitions, we state our main result of this section.

Theorem 5.1 *(Main result, covering number bound) If*

$$K \ge \max\left\{ \frac{X_b^2 B_b^2}{\epsilon^2}, \frac{n X_b^2 B_b^2}{\lambda_{\min}(\tilde{X}^T \tilde{X}) \min_{l=1,..,L} \frac{\delta_l}{\sum_{j=1}^p |\tilde{c}_{jl}|}} \right\},$$

$$\textit{then} \sup_{S \sim (\mu_{\mathcal{X}})^n} N(\sqrt{n}\epsilon, \mathcal{F}_{|S}, \|\cdot\|_2) \le \textit{count}(P_c^K). \quad (14)$$

The theorem above gives us a bound on the ℓ_2 covering number for the specially constrained function class $\mathcal{F}$.

Our new result is more in the spirit of [3], whose result makes use of Maurey's Lemma [2]. The main ideas of Maurey's Lemma are used in many machine learning papers in various contexts (*e.g.*, [16, 17, 18]).

The full proof with supporting lemmas as well as the question of computing the value of $count(P_c^K)$ is addressed in the full version [9]. Theorem 5.1 shows that we can limit the covering number by limiting operational cost, leading to a smaller hypothesis space, and a better generalization bound.

6 Conclusion

The perspective taken in this work contrasts with classical (and non-classical) statistics and machine learning; in those fields, prediction is often the only end goal. Our goal involves also how the model is used. There are many possible scenarios where including operational costs in statistical modeling could be very useful. In particular, this occurs when data are scarce or noisy, when the dimensionality is large and there is a lot of uncertainty in the model predictions, and when the operational cost has a steep gradient near the minimizer of the regularized loss. In our work, regularization terms involve optimization problems, not simply vector norms. We presented several example applications where including the operational cost substantially influenced the quality of the solution. Constraints on the operational costs lead to new types of hypothesis spaces, and we have obtained generalization bounds for a new type of hypothesis space involving arbitrary linear constraints.

Acknowledgements

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

[1] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[2] A.R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *Information Theory, IEEE Transactions on*, 39(3):930–945, 1993.

[3] Tong Zhang. Covering number bounds of certain regularized linear function classes. *Journal of Machine Learning Research*, 2:527–550, 2002.

[4] 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.

[5] John Ashworth Nelder and Roger Mead. A simplex method for function minimization. *Computer Journal*, 7(4):308–313, 1965.

[6] 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.

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

[8] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[9] Theja Tulabandhula and Cynthia Rudin. Machine learning with operational costs. Unpublished manuscript available on ArXiv at http://arxiv.org/abs/1112.0698, 2011.

[10] Theja Tulabandhula, Cynthia Rudin, and Patrick Jaillet. Machine learning and the traveling repairman. Unpublished manuscript available on ArXiv at http://arxiv.org/abs/1104.5061, 2011.

[11] Dimitris Bertsimas, David B. Brown, and Constantine Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, 2011.

[12] A. Ben-Tal, L. El Ghaoui, and A.S. Nemirovskiĭ. *Robust optimization*. Princeton Series in Applied Mathematics. Princeton University Press, 2009.

[13] A. Ben-Tal, D. Bertsimas, and D.B. Brown. A soft robust model for optimization under ambiguity. *Operations Research*, 2009.

[14] Peter L. Bartlett and Shahar Mendelson. Gaussian and Rademacher complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

[15] 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.

[16] V. Koltchinskii and D. Panchenko. Complexities of convex combinations and bounding the generalization error in classification. *The Annals of Statistics*, 33(4):1455–1496, 2005.

[17] R.E. Schapire, Y. Freund, P. Bartlett, and W.S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of statistics*, pages 1651–1686, 1998.

[18] 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.