Machine Learning and the Traveling Repairman

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

Patrick Jaillet JAILLET@MIT.EDU
Laboratory for Information and Decision Systems, Department of Electrical Engineering and Computer Science, and Operations Research Center
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

Abstract

The goal of the *Machine Learning and Traveling Repairman Problem* (ML&TRP) is to determine a route for a "repair crew," which repairs nodes on a graph. The repair crew aims to minimize the cost of failures at the nodes, but as in many real situations, the failure probabilities are not known and must be estimated. If there is uncertainty in the failure probability estimates, we take this uncertainty into account in an unusual way; from the set of acceptable models, we choose the model that has the lowest cost of applying it to the subsequent routing task. In a sense, this procedure agrees with a managerial goal, which is to show that the data can support choosing a low-cost solution.

Keywords: machine learning, traveling repairman, mixed-integer programming, uncertainty, generalization bound, constrained linear function classes

1. Introduction

We consider the problem of determining a route for a "repair crew" on a graph, where each node on the graph has some probability of failure. These probabilities are not known and must be estimated from past failure data. Intuitively the nodes that are more prone to failure should be repaired first. But if those nodes are far away from each other, the extra time spent by the repair crew traveling between nodes might actually increase the chance of failures occurring at nodes that have not yet been repaired. In that sense, it is better to construct the route to minimize the possible cost of failures, taking into account the travel time between nodes and also the (estimated) failure probabilities at each of the nodes. We call this problem the *machine learning and traveling repairman problem* (ML&TRP), and in this work, we present systematic approaches to formulating and solving this problem. There are many possible applications of the ML&TRP, including the scheduling of safety inspections or repair work for the electrical grid, oil rigs, underground mining, machines in a factory, or airlines.

One key idea of this work concerns the way uncertainty is handled in probabilistic modeling, and the way it relates to how the models are used in applications. Namely, when there is uncertainty in modeling, our idea is to choose a model that has advantages for our specific application, when we act on the predictions made by the model. Uncertainty in statistical modeling arises because the sample is finite, and there may be many predictive models that are equally good. If this is the case, in standard statistical and machine learning practice, we simply choose one of these models. When we do it this way, the algorithm choosing the model is oblivious to the way the model will be used in the application. Our idea is to choose a model that predicts well, and that also has the advantage of a low "operating cost," which is the cost to act on the predictions made by the model. For the ML&TRP, among all equally good predictive models for failure probabilities at the nodes, we choose the one that leads to the lowest failure cost. Incorporating the operating cost into the modeling can have substantial benefit, particularly when a small change in the model (one that may not affect prediction quality) leads to a large change in operating cost. This indeed can be true for the ML&TRP, as we will demonstrate.

We present two formulations for the ML&TRP, where one of them uses the new way of handling uncertainty. The first formulation is *sequential*: the failure probabilities are estimated in a way that is oblivious to the failure cost; then the route is determined by minimizing failure cost (which depends on the chosen probabilistic model). The second formulation handles uncertainty as discussed above, by computing the failure probabilities and the route *simultaneously*. This means that the estimated failure probabilities and the route are chosen together in a way that the failure cost will be low if possible; when there is uncertainty, the simultaneous formulation chooses the model with the lowest failure cost.

From an optimization perspective, there is a managerial reason to find low-cost solutions. A company might wish to know whether it is at all possible that a low-cost route can be designed, where the operational costs are realistically supported by the data; the simultaneous formulation finds such a solution. This type of formulation is optimistic; it provides the best possible (but still reasonable) scenario described by the data.

We present two possible choices for the failure cost, where either can be used for the sequential or the simultaneous formulations. The first failure cost is proportional to the sum (over nodes) of the expected number of failures at each node. The second failure cost considers, for each node, the probability that the first failure is before the repair crew's visit to the node. The first cost applies when the failure probability of a node does not change until it is visited by the crew, regardless of whether a failure already occurred at that node, and the second cost applies when the node is completely repaired after the first failure, or when it is visited by the repair crew, whichever comes first. In either case, the failure cost reduces to a weighted *traveling repairman problem* (TRP) objective, also called a minimum latency problem, or more generally, a time-dependent traveling salesman problem (see for instance, Picard and Queyranne, 1978).

Our method of handling uncertainty can be considered as a type of regularization, where we regularize by the operating cost. From a learning theory viewpoint, regularization (of any kind) limits the complexity of the hypothesis space used for the probabilistic model. This means that the failure cost term may assist with generalization, that is, the probabilistic model's ability to predict well on data drawn from the same distribution. In this work we present a generalization bound showing how a limitation on the failure cost might lead to a more accurate model for failure probabilities. There are two types of error in learning: estimation error and approximation error. The regularization term helps to control the estimation error, but will only reduce the approximation error if there really are low-cost solutions. For our application to the New York City power grid discussed below, there is a prior belief that probabilities that cannot be well-estimated will be low, implying

that the cost will also be low. Thus, we expect that incorporating the operating costs will help not only with estimation error, but also approximation error.

We will discuss a motivating application for the ML&TRP in the remainder of the introduction. In Section 2 we will outline the two formulations, and provide the two ways of modeling failure cost. In Section 3 we provide mixed-integer nonlinear programs (MINLP) and algorithms for solving the ML&TRP. Section 4 gives relevant illustrations, along with some experiments on data from the NYC power grid. Section 5 contains the theoretical generalization result. In Section 6 we discuss related literature and finally conclude in Section 7. The conference version of this work (Tulabandhula et al., 2011) contains a summary of our work on the ML&TRP. The ML&TRP is a new machine learning problem, meaning that as far as we know, there are no previous approaches to solving this particular problem in the literature. The natural approach is simply to solve the ML and TRP parts separately as in the sequential method, so we present this as a baseline. The TRP part alone has not been exactly addressed in the literature, since it is a particular weighted version of the standard TRP. The ML&TRP requires historical data for prediction, graph (distance) information, and the need for a route; this combination of data and routing could potentially be made available for a wide range of applications. Our particular interest is in a particular application that we now discuss.

Motivation

One particularly motivating application for the ML&TRP is smart grid maintenance. Since 2004, many power utility companies are implementing new inspection and repair programs for preemptive maintenance, whereas in the past, all repair work was done reactively (Urbina, 2004). An example of this is *vented manhole cover replacement programs*, where each manhole (which is an access point to the underground electrical network) in a city is replaced with a vented cover that allows gases to escape, mitigating the possibility and effects of serious events including fires and explosions.

New York City's power company Con Edison, which has such a replacement program, services tens of thousands of manholes in each borough, and it is not sensible for a repair crew to travel across the city and back again for each cover replacement. The scheduling of manhole inspection and repair work in Manhattan, Brooklyn and the Bronx is assisted by a machine learning model that estimates the probability of failure for each manhole within a given year (Rudin et al., 2010). Features for the model are derived from physical characteristics of the manhole (e.g., number of cables entering the manhole), and features derived from its history of involvement in past events. Repeat failures (serious and non-serious events) can occur on the same manhole. That said, failures are rare events, and it is not easy to accurately estimate the probability that a given manhole will fail within a given period of time. The current model for estimating failures does not take into account the route of the repair crew that replaces the covers. This leaves open the possibility that, for this domain and for many other domains, estimating the failure probabilities with knowledge of the route optimization procedure could lead to an improvement in repair operations.

The features for the NYC machine learning models are recomputed periodically, but not often, due to the expense of processing the raw data, and the fact that these probabilities change very slowly with time. Because of this, the route must be determined before the work starts. Also, the probabilities are not smooth along the graph, as discussed by Rudin et al. (2010); in initial attempts to model these probabilities on the Manhattan power grid, estimates were smooth geographically, and this type of model did not perform nearly as well as a more targeted model. In Manhattan it is very common to have relatively vulnerable manholes right next to manholes that are not vulnerable.

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

2. ML&TRP Formulations

In the ML&TRP, we are given two sets of instances, $\{x_i\}_{i=1}^m$, $\{\tilde{x}_i\}_{i=1}^M$, with $x_i \in \mathcal{X}$, $\tilde{x}_i \in \mathcal{X}$ that are feature vectors with $\mathcal{X} \subset \mathbb{R}^d$. Let x_i^j indicate the j-th coordinate of the feature vector x_i. For the first set of instances, we are also given labels $\{y_i\}_{i=1}^m$, $y_i \in \{-1, 1\}$. These instances and their labels form the training data. For the maintenance application, each of the $\{x_i\}_{i=1}^m$ encode manhole information (e.g., number and types of cables, number and types of previous events, etc.) and the labels $\{y_i\}_{i=1}^m$ encode whether the manhole failed ($y_i = 1$) or not ($y_i = -1$). More details about the features and labels can be found in Section 4. The other instances $\{\tilde{x}_i\}_{i=1}^M$ (with M unrelated to m), are unlabeled data that are each associated with a node on a graph G. The nodes of the graph G indexed by $i = 1, ..., M$ represent manholes on which we want to design a route. We are also given physical distances $d_{i,j} \in \mathbb{R}_+$ between each pair of nodes i and j. A route on G is represented by a permutation π of the node indices $1, \ldots, M$. Let Π be the set of all permutations of $\{1, ..., M\}$. Failure probabilities will be estimated at each of the nodes and these estimates will be based on a function of the form $f_\lambda(x) = \lambda \cdot x$. The class of possible functions $\mathcal{F}$ is chosen to be:

$$\mathcal{F} := \{f_\lambda : \lambda \in \mathbb{R}^d, \|\lambda\|_2 \leq M_1\}, \tag{1}$$

where M_1 is a fixed positive real number. As usual, it is possible to include an intercept by appending each instance x by a feature that is always 1 to hide the intercept.

The sequential formulation for the ML&TRP has a machine learning (ML) step and a traveling repairman (TRP) step. The simultaneous formulation has both the machine learning and traveling repairman together in the first step, and the second step chooses the route from the first step.

Sequential Formulation

Step 1. (ML) Compute the values $f_\lambda^*(\tilde{x}_i)$:

$$f_\lambda^* \quad \in \quad \text{argmin}_{f_\lambda \in \mathcal{F}} \text{LearningError}(f_\lambda, \{x_i, y_i\}_{i=1}^m).$$

Step 2. (TRP) Compute a route using estimated probabilities from f_λ^*:

$$\pi^* \in \text{argmin}_{\pi \in \Pi} \text{FailureCost}(\pi, f_\lambda^*, \{\tilde{x}_i\}_{i=1}^M, \{d_{i,j}\}_{i,j=1}^M).$$

The LearningError and FailureCost objectives will be defined shortly. The result $\pi^* \in \Pi$ is the route used for the repair crew. In Step 1, a transformation of $f_\lambda^*(x)$ yields an estimate of probability of failure $P(y = 1|x)$; this is the probability that a failure occurs in any given time step. We assume that the probability of failure is the same at each time step until something happens (either the crew visits, or a failure occurs), and that x does not change over the time that the route is being traversed. To ensure that these probabilities are in agreement with past observations, we choose $f_\lambda^*(x)$ to minimize a learning error in Step 1. In the second step, the route is chosen to minimize a weighted TRP cost based on those estimated probabilities. The sequential formulation is easier to solve than the simultaneous formulation outlined below.

Simultaneous Formulation

Step 1. Compute the values $f^*_\lambda(\tilde{x}_i)$:

$$f^*_\lambda \in \operatorname{argmin}_{f_\lambda \in \mathcal{F}} \Big[\text{LearningError}(f_\lambda, \{x_i, y_i\}_{i=1}^m) + C_1 \min_{\pi \in \Pi} \text{FailureCost}\left(\pi, f_\lambda, \{\tilde{x}_i\}_{i=1}^M, \{d_{i,j}\}_{i,j=1}^M\right) \Big].$$

Step 2. Compute a route corresponding to the scores:

$$\pi^* \in \operatorname{argmin}_{\pi \in \Pi} \text{FailureCost}(\pi, f^*_\lambda, \{\tilde{x}_i\}_{i=1}^M, \{d_{i,j}\}_{i,j=1}^M).$$

The result $\pi^* \in \Pi$ is the route used for the repair crew. In Step 1, f^*_λ is chosen to yield probability estimates that agree with the training data, but at the same time, yield lower failure costs. The user-defined constant C_1 is a tradeoff parameter, moving from "oblivious" estimation models to cost-aware estimation models. When C_1 is small, the algorithm essentially becomes sequential, ignoring the FailureCost. When it is large, the algorithm is highly biased towards low FailureCost solutions. One might want to choose C_1 large when there is a lot of uncertainty in the estimates and a strong belief that a very low cost solution exists. Or, one could choose a large C_1 to determine what policy would be chosen when the cost is underestimated. A small C_1 is appropriate when the number of training instances is large enough so that there is little flexibility (uncertainty) in the choice of model f^*_λ. Or one would choose low C_1 when we wish to choose, among equally good solutions, the one with the lowest cost.

We now define the LearningError and two options for the FailureCost.

2.1 LearningError

The unregularized error is a sum of losses over the training instances:

$$\sum_{i=1}^m l(f_\lambda(x_i), y_i),$$

where the loss function $l(\cdot, \cdot)$ can be any monotonically decreasing function bounded below by zero. We choose the logistic loss: $l(f_\lambda(x), y) := \ln\left(1 + e^{-y f_\lambda(x)}\right)$ so that the probability of failure $P(y = 1|x)$, is estimated as in logistic regression by:

$$P(y = 1|x) \text{ or } p(x) := \frac{1}{1 + e^{-f_\lambda(x)}}. \tag{2}$$

The negative log likelihood is the unregularized error:

$$-\text{log likelihood } = \sum_{i=1}^m -\ln\left[p(x_i)^{(1+y_i)/2}(1 - p(x_i))^{(1-y_i)/2}\right] = \sum_{i=1}^m \ln\left(1 + e^{-y_i f_\lambda(x_i)}\right).$$

We then add an ℓ_2 penalty over the parameters λ (with coefficient C_2) to get:

$$\text{LearningError}(f_\lambda, \{x_i, y_i\}_{i=1}^m) := \sum_{i=1}^m \ln\left(1 + e^{-y_i f_\lambda(x_i)}\right) + C_2 \|\lambda\|_2^2. \tag{3}$$

The coefficient C_2 is inversely related to the constant M_1 in (1) and both represent the same constraint on the function class. C_2 is useful for algorithm implementations whereas M_1 is useful for analysis.

2.2 Two Options for the FailureCost

The failure cost can be defined to match the application. We present two options here. In the first option (denoted as Cost 1), for each node there is a cost for (possibly repeated) failures prior to a visit by the repair crew. In the second option (denoted as Cost 2), for each node, there is a cost for the first failure prior to visiting it. There is a natural interpretation of the failures as being generated by a continuous random process at each of the nodes. When discretized in time, this is approximated by a Bernoulli process with parameter $p(\tilde{x}_i)$. Both Cost 1 and Cost 2 are appropriate for power grid applications. Cost 2 is also appropriate for delivery truck routing applications, where perishable items can fail (once an item has spoiled, it cannot spoil again). For many applications, neither of these two costs apply, in which case, it is possible to design a more appropriate or specialized cost and use that in place of the two we present here, using the same general idea of combining this cost with the learning error to produce an algorithm.

For convenience, we assume that after the repair crew visits all the nodes, it returns to the starting node (node 1) which is fixed beforehand. Scenarios where one is not interested in beginning from or returning to the starting node would be modeled slightly differently (the computational complexity remains about the same). For instance if we wanted to find an optimal tour route without specifying the starting node, we could choose each node in turn to be the starting node, solve our formulation with the fixed starting node M times, and pick the best of the M solutions.

Let a route be represented by $\pi : \{1, ..., M\} \mapsto \{1, ..., M\}$, this means that $\pi(i)$ is the i^{th} node to be visited. For example, let $M = 4, \pi = [2, 3, 4, 1]$. This means, $\pi(1) = 2$, node 2 is the first node to be visited, $\pi(2) = 3$, node 3 is the second node on the route, and so on. Let the distances be scaled appropriately so that a unit of distance is traversed in a unit of time. Given a route, the *latency* of a node $\pi(i)$ is the time (or equivalently distance) from the start at which node $\pi(i)$ is visited. It is the sum of distances traversed before position i on the route:

$$L_\pi(\pi(i)) := \text{time at which node } \pi(i) \text{ is visited} = \begin{cases} \sum_{k=1}^{M} d_{\pi(k)\pi(k+1)} \mathbf{1}_{[k<i]} & i = 2, ..., M \\ \sum_{k=1}^{M} d_{\pi(k)\pi(k+1)} & i = 1. \end{cases} \tag{4}$$

The assumption that the final node is the first node means that $d_{\pi(M)\pi(M+1)} = d_{\pi(M)\pi(1)}$. The starting node $\pi(1)$ thus has a latency $L_\pi(\pi(1))$ which is the total length of the route starting at node $\pi(1)$ and ending at node $\pi(1)$ after visiting all other nodes.

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

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

$$\begin{aligned} \text{Cost of node } \pi(i) &\propto E(\text{number failures in } L_\pi(\pi(i)) \text{ time units}) \\ &= \text{mean of } \text{Bin}(L_\pi(\pi(i)), p(\tilde{x}_{\pi(i)})) = p(\tilde{x}_{\pi(i)}) L_\pi(\pi(i)). \end{aligned} \tag{5}$$

Using this cost, if the failure probability for node $\pi(i)$, namely $p(\tilde{x}_{\pi(i)})$, is small, we can afford to visit it later on in the route which leads to a larger value of the latency $L_\pi(\pi(i))$. If $p(\tilde{x}_{\pi(i)})$ is large, we should visit node $\pi(i)$ earlier to keep our overall failure cost low.

The failure cost of route π is:

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

Substituting the definition of $L_\pi(\pi(i))$ from (4):

$$\text{FailureCost}(\pi, f_\lambda, \{\tilde{x}_i\}_{i=1}^M, \{d_{i,j}\}_{i,j=1}^M) = \sum_{i=2}^M p(\tilde{x}_{\pi(i)}) \sum_{k=1}^M d_{\pi(k)\pi(k+1)} \mathbf{1}_{[k<i]} + p(\tilde{x}_{\pi(1)}) \sum_{k=1}^M d_{\pi(k)\pi(k+1)}, \tag{6}$$

where $p(\tilde{x}_{\pi(i)})$ is given in (2). This will be Cost 1.

There are ways to make Cost 1 more general. The individual node cost in (5) assumes that the node's failure probability $p(\tilde{x}_{\pi(i)})$ becomes zero after the repair crew's visit, so that for the remainder of the route, the cost incurred at this node is $\propto 0 \times (L_\pi(\pi(1)) - L_\pi(\pi(i))$. We could relax this by assuming $p(\tilde{x}_{\pi(i)})$ does not vanish after the repair crew's visit and adding an additional cost for the expected failures in this period. That is, if β is a constant of proportionality for the cost after visiting node $\pi(i)$, then the cost would become:

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

If $\beta = 1$, then the repair crew does not have any effect and cost of each node is independent of its expected number of failures before the repair crew's visit. Typically, we expect that the repair crew will repair the node so that it will not fail, and the second term above is much larger than the first. Taking the constant of proportionality as $\beta = 0$, we return to the individual costs given by (5).

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

$$\text{FailureCost}(\pi, f_\lambda, \{\tilde{x}_i\}_{i=1}^M, \{d_{i,j}\}_{i,j=1}^M) = \sum_{i=1}^M p(\tilde{x}_i) L_\pi(i), \tag{7}$$

where $p(\tilde{x}_{\pi(i)})$ is given in (2).

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

This cost reflects the penalty for not visiting a node before the first failure occurs there. This model is governed by the geometric distribution: the probability that the first failure for node $\pi(i)$ occurs at time $L_\pi(\pi(i))$ is $p(\tilde{x}_{\pi(i)})(1 - p(\tilde{x}_{\pi(i)}))^{L_\pi(\pi(i))-1}$, and, substituting the expression (2) for $p(\tilde{x}_{\pi(i)})$, we have:

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

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

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

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

$$\sum_{i=1}^{M} \left(1 - \left(1 + e^{f_\lambda(\tilde{x}_{\pi(i)})}\right)^{-L_\pi(\pi(i))}\right). \tag{8}$$

This cost is not directly related to a weighted TRP cost in its present form. That is, when the failure probabilities of the nodes are all the same, the total cost is not linear in the latencies, as is the case for Cost 1. Building on this cost, we will derive a cost that is the same as a weighted TRP in Section 3.2, choosing it to be of the form:

$$\text{Cost of node } \pi(i) \quad \propto \quad L_\pi(\pi(i)) \log\left(1 + e^{f_\lambda(\tilde{x}_{\pi(i)})}\right), \tag{9}$$

as an alternative to (8).

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

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

If $\beta = 1$, then the sum above is 1 for all nodes regardless of node failures or latencies. More realistically, the cost of visiting the node after the failure is more than the cost of visiting proactively, $\beta \ll 1$ leading to (8).

We could again have written the summation to hide the dependence on π:

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

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

3. Optimization

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

3.1 Mixed-integer optimization for Cost 1

For either the sequential or simultaneous formulations, we need the solution of the subproblem:

$$\begin{aligned} \pi^* \quad &\in \quad \text{argmin}_{\pi\in\Pi} \text{FailureCost}(\pi, f_\lambda^*, \{\tilde{x}_i\}_{i=1}^M, \{d_{i,j}\}_{i,j=1}^M), \\ &= \quad \text{argmin}_{\pi\in\Pi} \sum_{i=2}^{M} p(\tilde{x}_{\pi(i)}) \sum_{k=1}^{M} d_{\pi(k)\pi(k+1)} \mathbf{1}_{[k<i]} + p(\tilde{x}_{\pi(1)}) \sum_{k=1}^{M} d_{\pi(k)\pi(k+1)}. \end{aligned} \tag{10}$$

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

$$\pi^* \in \text{argmin}_{\pi\in\Pi} \sum_{k=1}^{M} d_{\pi(k)\pi(k+1)}(M + 1 - k). \tag{11}$$

The standard TRP objective (11) is a special case of weighted TRP (10) when $\forall i = 1, ..., M,\ p(\tilde{x}_i) = p$:

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

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

We extend the standard TRP to include "unequal flow values" that will accommodate the more general problem (10). We use as a starting point the work of Fischetti et al. (1993) who give an integer programming formulation of the standard TRP. (Note that there are usually many ways that an integer program can be constructed, see Méndez-Díaz et al., 2008). The weights $\{\bar{p}(\tilde{x}_i)\}_i$ within the formulation below will be defined later. For interpretation, consider the sum of the probabilities $\sum_{i=1}^{M} \bar{p}(\tilde{x}_i)$ as the total "flow" through a route. At the beginning of the tour, the repair crew has flow $\sum_{i=1}^{M} \bar{p}(\tilde{x}_i)$. Along the tour, flow of the amount $\bar{p}(\tilde{x}_i)$ is dropped when the repair crew visits node $\pi(i)$ at latency $L_\pi(\pi(i))$. In this way, the amount of flow during the tour is the sum of the probabilities $\bar{p}(\tilde{x}_i)$ for nodes that the repair crew has not yet visited. We introduce two sets of variables $\{z_{i,j}\}_{i,j}$ and $\{y_{i,j}\}_{i,j}$ that together represent a route (instead of the π notation). Let $z_{i,j}$ represent the flow on edge (i, j) and let a binary variable $y_{i,j}$ represent whether there exists a flow on edge (i, j). (There will only be a flow along the route, and there will not be a

flow along edges that are not in the route.) The mixed-integer program is as follows:

$$\min_{z,y} \sum_{i=1}^{M}\sum_{j=1}^{M} d_{i,j} z_{i,j} \quad \text{s.t.} \tag{12}$$

$$\text{No flow from node } i \text{ to itself: } \; z_{i,i} = 0 \;\; \forall i = 1, ..., M \tag{13}$$

$$\text{No edge from node } i \text{ to itself: } \; y_{i,i} = 0 \;\; \forall i = 1, ..., M \tag{14}$$

$$\text{Exactly one edge into each node: } \; \sum_{i=1}^{M} y_{i,j} = 1 \;\; \forall j = 1, ..., M \tag{15}$$

$$\text{Exactly one edge out from each node: } \; \sum_{j=1}^{M} y_{i,j} = 1 \;\; \forall i = 1, ..., M \tag{16}$$

$$\text{Flow coming back to the initial point at the end of the loop is } \bar{p}(\tilde{x}_1)\text{: } \; \sum_{i=1}^{M} z_{i,1} = \bar{p}(\tilde{x}_1) \tag{17}$$

Change of flow after crossing node k is either $\bar{p}(\tilde{x}_k)$ or it is $\bar{p}(\tilde{x}_1)$ minus the sum of $\bar{p}$'s:

$$\sum_{i=1}^{M} z_{i,k} - \sum_{j=1}^{M} z_{k,j} = \begin{cases} \bar{p}(\tilde{x}_1) - \sum_{i=1}^{M} \bar{p}(\tilde{x}_i) & k = 1 \\ \bar{p}(\tilde{x}_k) & k = 2, ..., M \end{cases} \tag{18}$$

$$\text{Connects flows } z \text{ to indicators of edge } y\text{: } \; z_{i,j} \leq r_{i,j} y_{i,j} \tag{19}$$

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

Constraints (13) and (14) restrict self-loops from forming. Constraints (15) and (16) impose that every node should have exactly one edge coming in and one going out. Constraint (17) represents the flow on the last edge coming back to the starting node. Constraint (18) quantifies the flow change after traversing a node k. Constraint (19) represents an upper bound on $z_{i,j}$ relating it to the corresponding binary variable $y_{i,j}$.

For Cost 1, we define the weights $\bar{p}(\tilde{x}_i)$ to be equal to the estimated failure probabilities. That is, $\bar{p}(\tilde{x}_i) := 1/(1 + e^{-\lambda \cdot \tilde{x}_i})$.

3.2 Mixed integer optimization for Cost 2

Here we reason about the choice for changing Cost 2 in (8) to resemble (9). Starting with the sum (8) over costs (8),

$$\min_{\pi} \sum_{i=1}^{M} \left(1 - \left(1 + e^{f_\lambda(\tilde{x}_{\pi(i)})}\right)^{-L_\pi(\pi(i))}\right),$$

we apply the log function to the second term of the cost of each node (8) to get a new cost

$$\left(1 - \log\left(1 + e^{f_\lambda(\tilde{x}_{\pi(i)})}\right)^{-L_\pi(\pi(i))}\right),$$

and the minimization becomes instead:

$$\min_{\pi} \sum_{i=1}^{M} \left(1 - \log\left(1 + e^{f_\lambda(\tilde{x}_{\pi(i)})}\right)^{-L_\pi(\pi(i))}\right)$$

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

where the first term is the sum over nodes of the expression (9). This failure cost term is now a weighted sum of latencies where the weights are of the form $\log\left(1+e^{f_\lambda(\tilde{x}_{\pi(i)})}\right)$. We can thus reuse the mixed integer program (12)-(19) where the weights are redefined as $\bar{p}(\tilde{x}_i) := \log\left(1+e^{\lambda\cdot\tilde{x}_i}\right)$. Now that the TRP subproblem has been completely defined for both Cost 1 and Cost 2, we will discuss first how to solve the subproblem alone, which is Step 2 of the sequential method. Then we will discuss how solvers for the simultaneous method.

3.3 Solvers for the weighted TRP subproblem

A generic MILP solver like CPLEX[1] or Gurobi[2] can produce an exact solution using branch-and-bound or other related exact methods. In our experiments we use Gurobi. The weighted TRP problem is NP-hard (can be shown by a reduction to the hamiltonian cycle problem) and hence most likely not solvable by polynomial-time algorithms. The standard unweighted (all weights equal) TRP can be encoded by different mixed-integer programming formulations (see Fischetti et al., 1993; Eijl van, 1995; Méndez-Díaz et al., 2008) each with different performance guarantees (e.g., solving 15-60 nodes), which could be adapted for our purpose. There are also techniques for producing constant factor approximate solutions to the unweighted TRP, which could run faster than the MILP solvers for large problem instances. If the weights $\{w_i\}_i$ are integers, we can adapt these faster techniques for the standard problem to the weighed TRP problem by replicating each node w_i times. If the weights are rational, as is the case in (20) and (21), we can use rounding and discretization in order to apply the faster solution techniques for solving the standard TRP. More on this topic is discussed in Section 6.

3.4 Mixed-integer nonlinear programs (MINLPs)

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

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

$$
\text{constraints (13) to (19) hold, where } \bar{p}(\tilde{x}_i)=\frac{1}{1+e^{-\lambda\cdot\tilde{x}_i}},
$$

1. IBM ILOG CPLEX Optimization Studio v12.2.0.2 2010
2. Gurobi Optimizer v3.0, Gurobi Optimization, Inc. 2010

or equivalently,

$$\min_{\lambda} \left(\sum_{i=1}^{m} \ln\left(1 + e^{-y_i f_\lambda(x_i)}\right) + C_2\|\lambda\|_2^2 + C_1 \min_{\{z_{i,j}, y_{i,j}\}} \sum_{i=1}^{M}\sum_{j=1}^{M} d_{i,j} z_{i,j} \right) \quad \text{s.t.} \tag{20}$$
$$\text{constraints (13) to (19) hold, where } \bar{p}(\tilde{x}_i) = \frac{1}{1 + e^{-\lambda \cdot \tilde{x}_i}}.$$

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

$$\min_{\lambda} \left(\sum_{i=1}^{m} \ln\left(1 + e^{-y_i f_\lambda(x_i)}\right) + C_2\|\lambda\|_2^2 + C_1 \min_{\{z_{i,j}, y_{i,j}\}} \sum_{i=1}^{M}\sum_{j=1}^{M} d_{i,j} z_{i,j} \right) \quad \text{s.t.} \tag{21}$$
$$\text{constraints (13) to (19) hold, where } \bar{p}(\tilde{x}_i) = \log\left(1 + e^{\lambda \cdot \tilde{x}_i}\right).$$

If we have an algorithm for solving (20), then the same scheme can be used to solve (21). There are multiple ways of solving (or approximately solving) a mixed integer non-linear optimization problem of the form (20) or (21). We consider three methods here, described next. The first method is to directly use a generic mixed integer non-linear programming (MINLP) solver. The second and third methods are iterative schemes over the λ parameter space, called Nelder-Mead and Alternating Minimization, denoted NM and AM respectively. At every iteration of the NM and AM algorithms, we will need to evaluate the objective function. This evaluation involves solving an instance of the weighted TRP subproblem discussed in the previous subsections.

Method 1: MINLP Solver

For our experiments we directly use a MINLP solver called Bonmin (Bonami et al., 2008). These types of solvers typically use general MILP solving techniques like branch and bound or dynamic programming interleaved with continuous optimization. Since the general MILP solving techniques, as discussed, can take exponential time when applied directly to our formulations, the MINLP solvers which use them can in turn, be inefficient if the graph is moderate to large in size. However, when the graph is small, for instance when we want to schedule a tour over only a few nodes, the MINLP solver can generally compute a solution to the problems (20) or (21) in a manageable period of time.

Method 2: Nelder-Mead in λ-space (NM)

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

Algorithm 1 AM: Alternating minimization algorithm

Inputs: $\{x_i, y_i\}_1^m, \{\tilde{x}_i\}_1^M, \{d_{ij}\}_{ij}, C_1, C_2, T$ and initial vector λ_0.
for t=1:T **do**
  Compute $\pi_t \in \text{argmin}_{\pi\in\Pi}\text{Obj}(\lambda_{t-1}, \pi)$.
  Compute $\lambda_t \in \text{argmin}_{\lambda\in\mathbb{R}^d}\text{Obj}(\lambda, \pi_t)$.
end for
Output: π_T.

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

Define Obj as a function of λ and π:

$$\text{Obj}(\lambda, \pi) = \text{LearningError}(f_\lambda, \{x_i, y_i\}_{i=1}^m) + C_1\text{FailureCost}\left(\pi, f_\lambda, \{\tilde{x}_i\}_{i=1}^M, \{d_{i,j}\}_{i,j=1}^M\right). \quad (22)$$

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

4. Experiments

We have now defined two formulations (sequential and simultaneous), each with two possible definitions for the failure cost (Cost 1 and Cost 2), and three algorithms for the simultaneous formulation (MINLP solver, NM, and AM). In what follows, we will show how the simultaneous formulation takes advantage of uncertainty. In particular, we will provide circumstances where it is possible to produce low cost solutions while maintaining the same level of prediction quality. We will highlight the advantage of the simultaneous method over the less general sequential method, in that the sequential formulation will not necessarily yield the best result: the best possible minimizer of the learning error does not necessarily yield a low cost solution.

In each experiment, there is a fixed training set and separate test set to evaluate predictions of the model, and there is an unlabeled set of nodes with distances. In these experiments, there is a lot of uncertainty in the estimates for the unlabeled set in particular, so the probabilities could reasonably change without substantially affecting general prediction ability. For instance, in the experiment discussed below regarding maintenance on the New York City power grid, the data are very imbalanced (the positive class is very small), so there is a lot of uncertainty in the estimates, and further, there is a prior belief that a low-cost route exists. In particular, we have reason to believe that some of the probabilities are overestimated using the particular unlabeled set we chose. We also believe that knowing the repair route can help to determine these probabilities; this is because there are underground electrical cables traversing each linear stretch of the repair route.

4.1 Illustrative Experiments

We will show how a small change in the probabilities produced by the model can give a completely different route and cost. In the first illustration, the probabilities change slightly as a result of the failure cost term. In the second illustration, the probability estimates change dramatically due to the failure cost term, but still most of the estimates change by

a small amount. In both cases the unlabeled points are in a low-density region, giving rise to uncertainty in the estimates for those instances, and different routes.

Illustration I

We pick a graph G with distances $\{d_{i,j}\}_{i,j}$ as shown in Figure 1(a). The number of nodes, which is also equal to the number of unlabeled instances is $M = 4$. We also pick two sets of probability estimates, shown under the node labels. These sets of probability estimates are only slightly different from each other; Figure 1(a) shows the graph with the first set of probability estimates and Figure 1(b) shows the same graph with the other set of estimates. Even though the probability estimates are so similar, the optimal routes corresponding to the minimum failure cost (Cost 1) are entirely different. The route corresponding to the first set of probability values, shown in Figure 1(a), is $\pi^*= 1-3-2-4-1$. The route corresponding to the second set of probability values, shown in Figure 1(b), is $\pi^*= 1-3-4-2-1$.

The probability estimates of Figure 1(a) could easily be the result of the sequential formulation and those of Figure 1(b) could be the result of simultaneous formulation. To show one circumstance where this is possible, consider feature vectors in the plane, $\tilde{x}_1, \ldots, \tilde{x}_4$ in $\mathbb{R}^2$, as illustrated in Figure 2. The training instances are also shown in the same figure, represented by two gray clusters here. (Note that there are many ways the training instances could be distributed to lead to the same probability estimates, we illustrate only one such pattern.) In this feature space, the sequential formulation might produce a function whose 0.5-probability level set is displayed as a black line in Figure 2. This function determines the probability estimates at the four nodes. The simultaneous formulation produces a function with a slightly different 0.5-probability level set, for example shown by the dashed line in Figure 2. This setup would lead to probability estimates similar to those provided in Figure 1(a) and Figure 1(b) respectively. The second set of probability values (corresponding to simultaneous formulation) yields a substantially lower value of Cost 1 than the first set, in particular the failure cost decreases by $\sim 16.4\%$. So a solution from the simultaneous formulation would be preferred.

It is important not to confuse the feature space of x_i's (2-dimensional here) with the space that the graph $\{d_{i,j}\}_{i,j}$ is embedded in (also 2-dimensional here). These are different spaces in general, and the ML&TRP graph need not even have a physical distance interpretation.

Illustration II

We will show that a large change in the probability model does not necessarily lead to a large change in overall prediction accuracy, but may lead to a large change in route. The training set was chosen uniformly at random from a distribution that is uniform over two triangles pointing end to end. We used six unlabeled points as the nodes. Figure 3 shows the training instances and unlabeled instances in feature space along with two level sets. The first one, colored black, is the estimated level set for $P(y = 1|x) = 0.5$ learned from ℓ_2-regularized logistic regression. The second level set, colored red and also drawn at probability estimate 0.5, is learned from the new simultaneous formulation, with failure cost modeled according to Cost 1. Node 6 (triangle with label "unlabeled $\tilde{x}_i$" in Figure 3) lies in a low density region of feature space, so its probability cannot be well estimated. For the sequential formulation, node 6 was assigned $p(\tilde{x}_6) = 0.5$. For the sequential formulation, the optimal route obtained by solving the weighted TRP problem is $1-2-3-6-4-5-1$, shown in Figure 4. For the simultaneous formulation, node 6 has been assigned a new probability value $p(\tilde{x}_6) = 0.29$. This change is possible because node

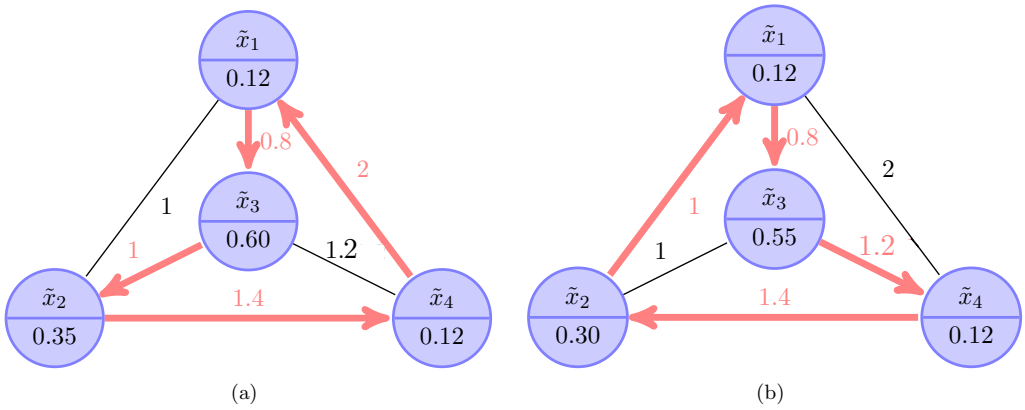


Figure 1: Physical space for the four node illustration. The numbers in the nodes indicate their probability of failure, and the numbers on the edges indicate distances. (a) Route as determined by the sequential formulation is highlighted. (b) Route determined by the simultaneous formulation.

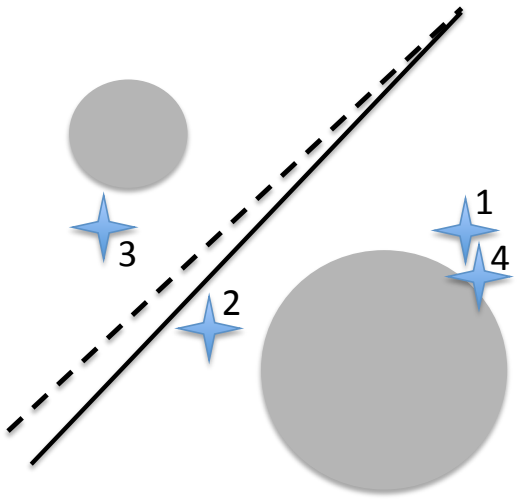


Figure 2: One possible feature space for Illustration I.

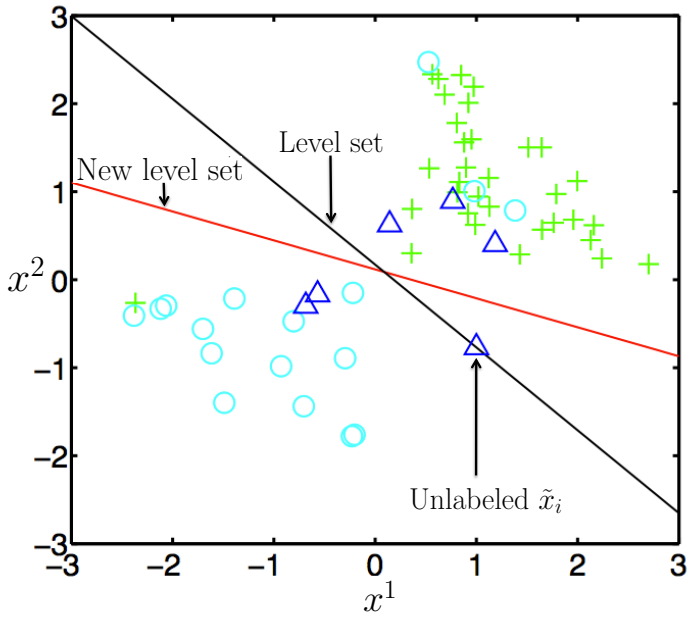


Figure 3: The two-dimensional feature space for Illustration II. x^1 and x^2 represent the first and second coordinates respectively. Solution models to the estimation problem are denoted by their 0.5 probability level sets. Black corresponds to the sequential and red corresponds to the simultaneous formulation. For both solutions, on one side of the line is the region of high probability of failure and on the other side is the region of low probability of failure. Both of them get the training data: '+'s and '∘'s, mostly on the correct sides. The triangles represent the unlabeled data $\{\tilde{x}_i\}_{i=1}^6$.

6's probability estimate can vary quite a lot without changing the probability estimates of others. This changes the route to $1-2-3-4-5-6-1$ as shown in Figure 5.

Node 6 is also physically far from all other nodes. If it has a high enough probability estimate compared to nodes 4 and 5 (blue triangles in the lower left half of Figure 3), then a route that visits node 6 before visiting nodes 4 and 5 would be favored; this is what happens in the sequential formulation. In the simultaneous formulation, we chose C_1 large enough so that the tour route visits 4 and 5 before 6. This results in a $\sim 9\%$ decrease in the failure cost (Cost 1), with a $\sim 3\%$ change in the learning error. In particular, for the sequential method, Cost 1 is 4.7 units and the learning error is 15.7 units; for the simultaneous method, Cost 1 is 4.25 units and the learning error is 16.2 units ($C_1 = 5 \times 10^{-4}$).

Since the feature space is two-dimensional in this illustration, λ will be three dimensional (when we include the intercept). Keeping the intercept fixed, we can plot the surface of both the learning error term and the failure cost term as a function of the two coordinates of λ, corresponding to the two features. The ℓ_2-regularized learning error is plotted in Figure 6(a). The optimal failure cost (Cost 1) was computed for each λ and is plotted

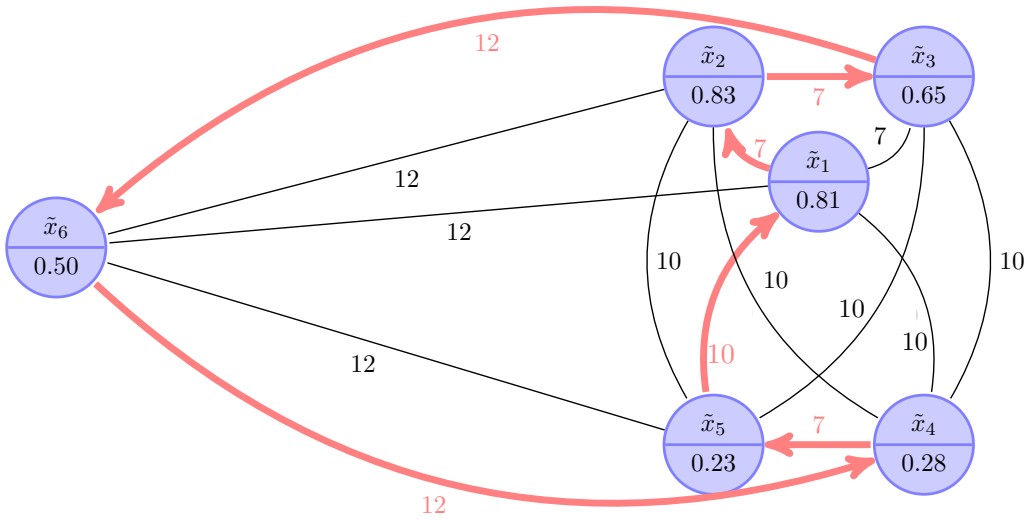


Figure 4: Physical space for the six node illustration, sequential formulation. The numbers in the nodes indicate their probability of failure, and the numbers on the edges indicate distances. The optimal route $1-2-3-6-4-5-1$ as determined by the sequential formulation is highlighted.

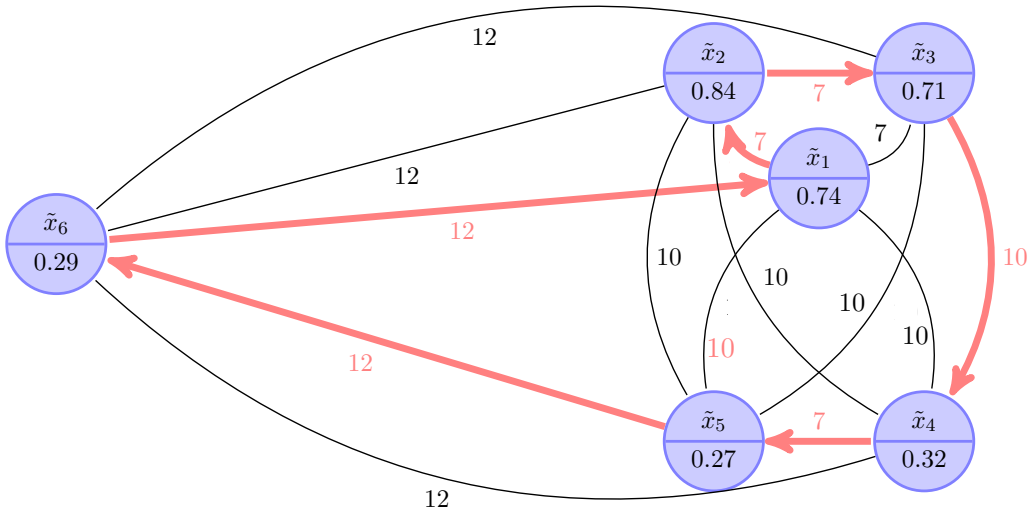
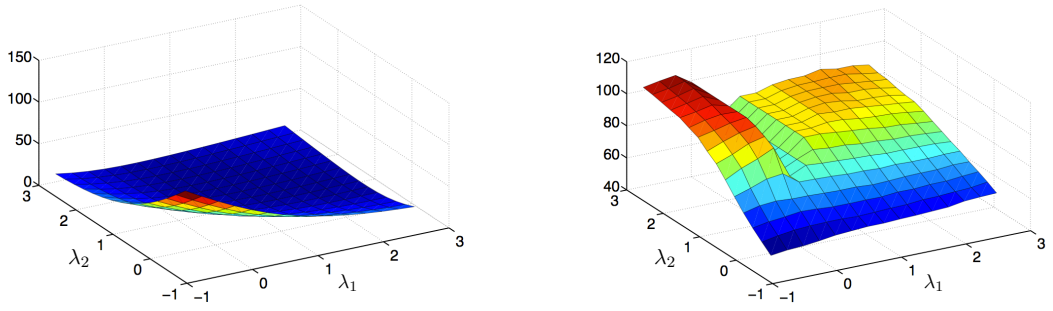


Figure 5: The optimal route $1-2-3-4-5-6-1$ as determined by the simultaneous formulation is highlighted.



(a) (b)

Figure 6: The values of the two terms in the objective of the simultaneous ML&TRP formulation. (a) Learning error as a function of $\{\lambda_1, \lambda_2\}$. The last coordinate, λ_3 is kept fixed. (b) Scaled optimal failure cost (Cost 1 divided by 100) over a 2D grid of λ_1 and λ_2, again with λ_3 fixed.

in Figure 6(b); to obtain each point on the surface, we needed to solve a weighted TRP subproblem. The simultaneous ML&TRP objective is the sum of the values in Figures 6(a) and 6(b), and the constant C_1 controls how these surfaces are added together. If the learning error term in Figure 6(a) is somewhat flat near the minimizer of the ML&TRP objective and the failure term in 6(b) is not flat, the failure term may be able to have a substantial effect on the solution. This is precisely the type of circumstance when the simultaneous formulation can impact the quality of the solution.

4.2 ML&TRP on the NYC power grid

We now illustrate the performance of our method on a data set obtained from a collaborative effort with Con Edison, which is NYC's power utility company. More details about these data can be found in (Rudin et al., 2010). This dataset was developed in order to assist Con Edison with its maintenance and repair programs on the secondary electrical distribution network in NYC; specifically, it was designed for the purpose of predicting manhole fires and explosions. We chose to use all manholes from the Bronx (∼23K manholes). Each manhole is represented by (4-dimensional) features that encode the number and type of electrical cables entering the manhole and the number and type of past events involving the manhole. The event features encode how often in the past the manhole was the source of partial outages, full outages and/or underground burnouts. The training features encode events prior to 2008, and the training labels are 1 if the manhole was the source of a serious event (fire, explosion, smoke) during 2008. The prediction task is to predict events in 2009. The test set (for evaluating the performance of the predictive model) consists of features derived from the time period before 2009, and labels from 2009. In our experiments, for both training and test we had a large sample (∼23K instances). Predicting manhole events can a difficult task for machine learning, because one cannot necessarily predict an event using the available data. The operational task was to design a route for a repair crew that is fixing seven manholes in 2009 on which we want the cost of failures to be low.

Manhole failures are rare events. This means there are many more negative labels than positive labels. Using a logistic model gives probability estimates which are low overall, so the misclassification error is almost always the size of the whole positive class. Because of this, we evaluate the quality of the predictions from f_{λ^*} using the area under the ROC curve (AUC), for both training and test.

The nodes are 7 randomly chosen manholes, and the features for the nodes encode events prior to 2009. The distances between the nodes were obtained from Google Maps, by querying the driving distance between each pair of nodes. Note that we do not want 'flying' distance between two coordinates as this can be very different from the actual driving distance.

We solved (20) and (21) for a range of values for the regularization parameter C_1, with the goal of seeing whether for the same level of estimation performance, we can get a reduction in the cost of failures. The evaluation metric, AUC, is a measure of ranking quality; it is sensitive to the rank-ordering of the nodes in terms of their probability to fail, and it is not as sensitive to changes in the values of these probabilities. This means that as the parameter C_1 increases, the estimated probability values will tend to decrease, and thus the failure cost will decrease; it may be possible for this to happen without impacting the prediction quality as measured by the AUC, but this is not guaranteed.

Results

The test AUC values for the simultaneous method were all within 1% of the values obtained by the sequential method; this is true for both Cost 1 and Cost 2, for each of the AM, NM, and MINLP solvers, see Figures 7 and 8. The variation in learning error across the methods was also small, about 2%, see Figure 9. So, changing C_1 did not dramatically impact the prediction quality as measured by the AUC. On the other hand, the failure costs varied widely over the different methods and settings of C_1, as a result of the decrease in the probability estimates, as shown in Figure 10. As C_1 was increased from 0.05 to 0.5, Cost 1 went from 27.5 units to 3.2 units, which is over eight times smaller. This means that with a 1-2% variation in the predictive model's AUC, the failure cost can decrease a lot, potentially yielding a more cost-effective route for inspection and/or repair work. The reason for an order of magnitude change in the failure cost is because the probability estimates are reducing by an order of magnitude due to uncertainty at the nodes; yet our model still maintained the same overall level of AUC performance on training and test sets.

The slight increase in learning error for the simultaneous formulation (using Cost 1) as a function of C_1 is shown also in Figure 9. The steep decrease in failure cost as a function of C_1 is shown in Figure 10. Note that accuracy can be increased in this problem by adding more features: a related work on the same dataset (Rudin et al., 2011) achieves about 10% increase in the AUC over what we report here.

In Figures 11-13 we show the routes according to the different algorithms. We first provide the naïve route in Figure 11, which was obtained by estimating probabilities using ℓ_2-penalized logistic regression, and then simply visiting nodes according to decreasing values of these probabilities. Figure 12 shows the route provided by the sequential formulation. For the simultaneous method, there are changes in the route as the coefficient C_1 increases. When C_1 is low, the route is the same as obtained from the sequential method, in Figure 12. When the failure term starts influencing the optimal solution of the objective (20) because of an increase in C_1, we get a new route, depicted in Figure 13. In most applications relevant to this problem, we suspect that the solution used in practice is somewhere in between the naïve route and the sequential route, in that a human views the naïve solution and adjusts it by hand to be closer to the sequential route (without solving the TRP). For the application to electrical grid maintenance, the simultaneous method was

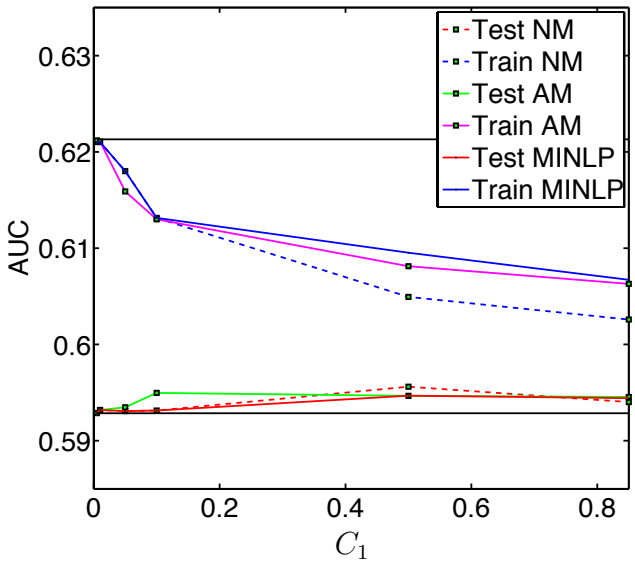


Figure 7: The AUC values corresponding to model parameters obtained from the simultaneous formulation using Cost 1 by NM-MILP and AM-MILP algorithms along with MINLP solver, plotted as a function of C_1. The AUC values on the training data decrease slightly and the same values for test data increase marginally. The two horizontal lines represent the training and test AUC values obtained by ℓ_2-penalized logistic regression and thus, are constant with respect to C_1.

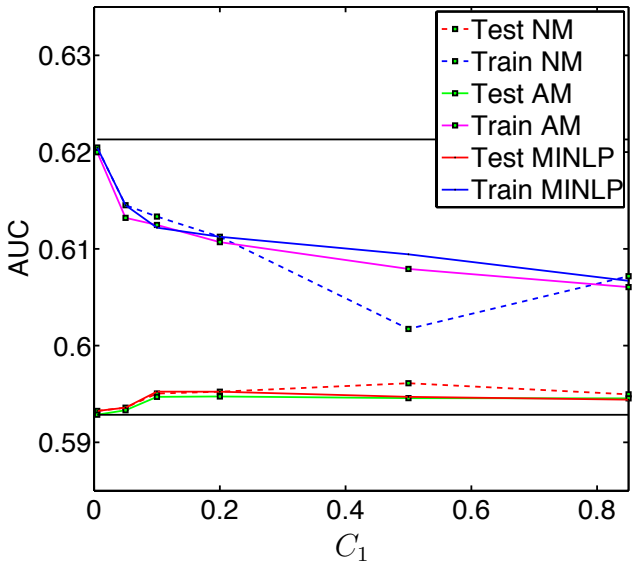


Figure 8: The AUC values obtained from the simultaneous formulation, using Cost 2, from the NM-MILP and AM-MILP algorithms along with the MINLP solver, plotted as a function of C_1. Again, the training data AUC values decrease and the test data AUC values remain nearly constant. The horizontal lines represent constant values of the AUC obtained by ℓ_2-penalized logistic regression.

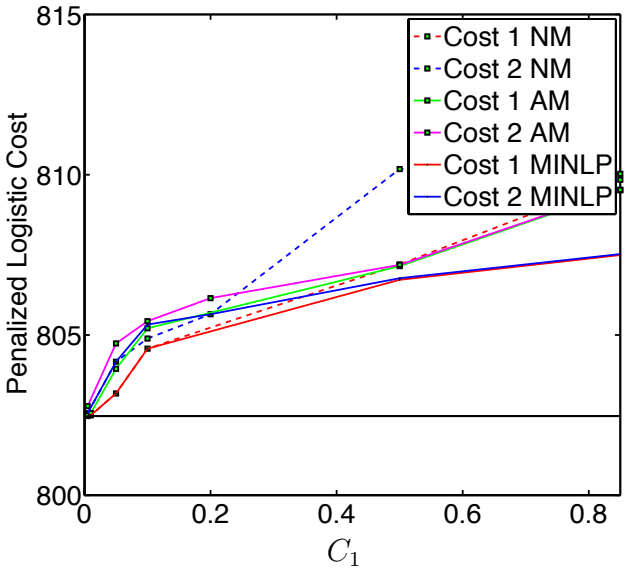


Figure 9: The ℓ_2-regularized logistic loss increases as a function of increasing C_1. The horizontal line represents the loss value from ℓ_2-penalized logistic regression with no regularization ($C_1 = 0$).

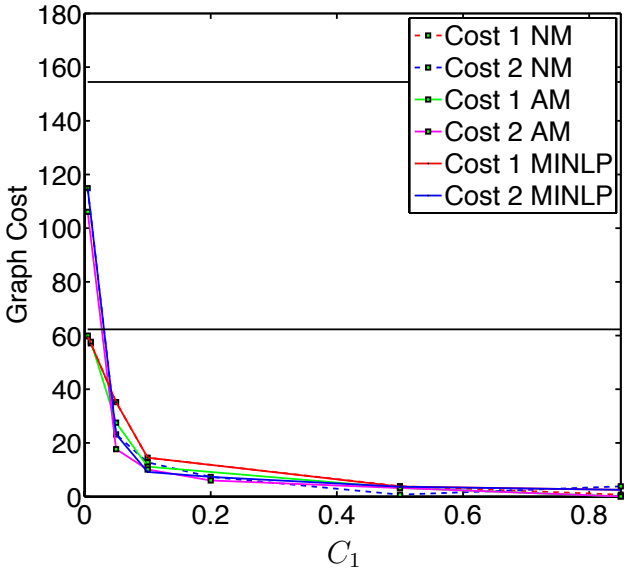


Figure 10: The failure costs decrease as a function of the regularization parameter C_1. The horizontal lines in the figure represent the sequential formulation solutions; the lower horizontal line is Cost 1 of the solution obtained by ℓ_2-penalized logistic regression, and the upper line is Cost 2 of that solution.

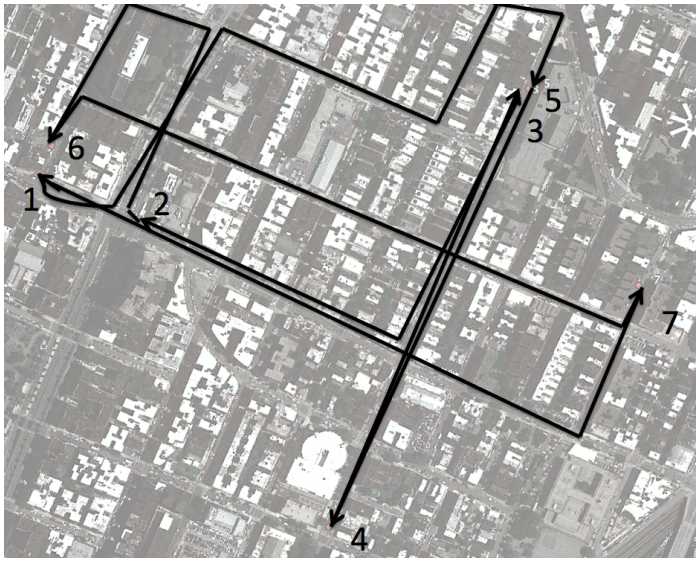


Figure 11: A naïve route: 1-5-4-3-2-6-7-1 obtained by sorting the probability estimates in decreasing order and visiting the corresponding nodes.

Figure 12: Sequential formulation route: 1-5-3-4-2-6-7-1. The simultaneous formulation also chooses this route when C_1 is small.

able to find a substantially lower cost route that the naïve or sequential method, with little (if any) change in the AUC prediction quality. This demonstration on data from the Bronx indicates that it is possible to take advantage of uncertainty in modeling in order to create a much more cost-effective solution.

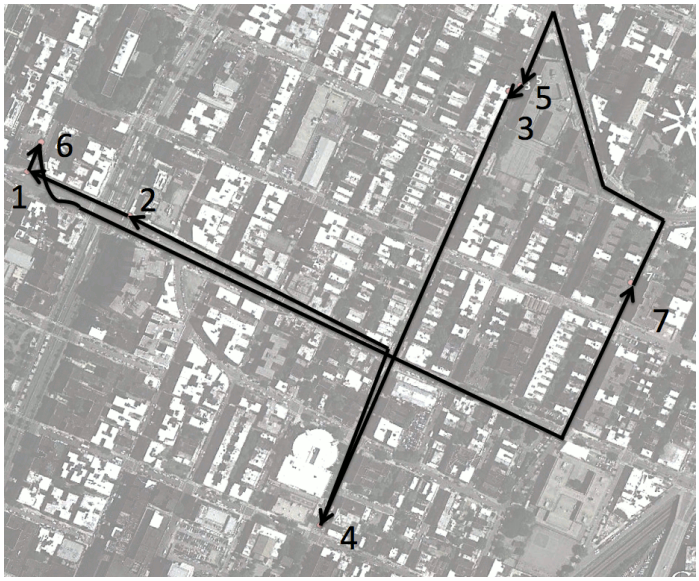


Figure 13: Route chosen by the simultaneous formulation when C_1 is larger: 1-6-7-5-3-4-2-1. Prediction performance is only slightly influenced by the route change, but the cost of the route (Cost 1) decreases a lot.

5. Generalization Bound

We initially introduced the failure cost regularization term in order to find scenarios where the data would support low-cost (more actionable) repair routes. From a learning theoretic point of view, incorporating regularization reduces the size of the hypothesis space and may thus promote generalization. The size of the hypothesis space can be controlled using C_1. Increasing C_1 may thus assist in predicting failure probabilities. In what follows, we will provide a generalization bound for the ML&TRP algorithm (20) with Cost 1. A similar bound for the formulation with Cost 2, (21), might be derived using the same method.

Generalization bounds are probabilistic guarantees that are useful for showing what variables may be important in the learning process (Bousquet, 2003). The vast majority of works on generalization analysis are mainly interested in problems where the dimensionality d of the input space (or feature space) is very large, leading to the "curse of dimensionality." In that case, various measures of the complexity of the hypothesis space are incorporated into the bounds; these complexity measures can often gauge the richness of a class of functions in a way that is independent of the input dimension d. Examples of such measures include the VC dimension for $\{0,1\}$-valued function classes, ϵ-fat shattering dimension for real valued functions, Rademacher complexity, and certain kinds of covering numbers (Vapnik, 1998; Bartlett and Mendelson, 2002; Mendelson and Vershynin, 2003; Zhang, 2002; Shawe-Taylor and Cristianini, 2002; Kolmogorov and Tikhomirov, 1959). In the present work, we are instead interested in how the failure cost influences generalization for a fixed d, that is, we are interested in how C_1 affects generalization, and not so much interested in the dependence on d. We make the assumption that all input features affect prediction ability. This means that our bound will depend on the dimensionality of the input space d. Such a dependence on d is not uncommon; for example, covering number bounds depending on the "Pollard dimension" (equal to input dimension d when finite)

have been obtained for bounded real-valued functions (see Theorem 14.21 of Anthony and Bartlett, 1999, Theorem 6 of Haussler, 1992). There are also many bounds that rely directly on the number of elements within the hypothesis space, for finite hypothesis spaces.

Note that better generalization does not in general necessarily imply a better model. Increasing C_1 increases the bias of the model and helps to reduce the variance, but any type of bias can either help or hurt the quality of a model, depending on whether the "prior belief" associated with the bias is correct, and leads to a decrease in approximation error. The prior belief in our case is that a low-cost model exists. This section deals only with the generalization error, not the approximation error.

Since generalization concerns with how close $f_\lambda(x)$ is to y when instance (x, y) is drawn from an unknown distribution $\mu_{\mathcal{X}\times\mathcal{Y}}$, this new instance x is unrelated to the graph we used to form our failure cost. So far as getting a low failure cost route on the unlabeled data was concerned, we achieved that objective by solving one of the MINLPs. We also learned f_λ in the process. The question we ask now is how good is this learned function f_λ on unseen instances. In order to do so, we seek to bound the true risk

$$R^{\text{true}}(f_\lambda) := E_{(x,y)\sim\mu_{\mathcal{X}\times\mathcal{Y}}} l(f_\lambda(x), y) = \int \ln\left(1 + e^{-yf_\lambda(x)}\right) \partial\mu_{\mathcal{X}\times\mathcal{Y}}(x, y),$$

where $l : f_\lambda(\mathcal{X}) \times \mathcal{Y} \to \mathbb{R}$, is chosen to be the logistic loss. We will bound $R^{\text{true}}(f_\lambda)$ by the empirical risk:

$$R^{\text{emp}}(f_\lambda, \{x_i, y_i\}_1^m) = \frac{1}{m}\sum_{i=1}^{m} l(f_\lambda(x_i), y_i) = \frac{1}{m}\sum_{i=1}^{m} \ln\left(1 + e^{-y_i f_\lambda(x_i)}\right)$$

plus a complexity term, which measures the capacity of the hypothesis space of functions of the form f_λ.

We have as usual that $f_\lambda \in \mathcal{F}$ where $\mathcal{F}$ corresponds to a ball of radius M_1 in λ-space (a ball in $\mathbb{R}^d$):

$$\mathcal{F} := \{f_\lambda : f_\lambda(x) = \lambda \cdot x, \lambda \in \mathbb{R}^d, \|\lambda\|_2 \le M_1\}.$$

Standard bounds consider the complexity of $\mathcal{F}$. However, the hypothesis space for the ML&TRP is smaller than $\mathcal{F}$, since we have also the constraint on the failure cost. Replacing the Lagrange multiplier C_1 with an explicit constraint on the failure cost (7), we have that for the ML&TRP, f_λ is subject to the failure cost constraint: $\min_\pi \sum_{i=1}^{M} p(\tilde{x}_{\pi(i)}) L_\pi(\pi(i)) \le C_{\text{budget}}$, where C_{budget} is inversely related to C_1, controlling a "budget" for the failure cost. Let us define the set of functions that are subject to a constraint on the failure cost, plugging in the form for $p(\tilde{x}_i)$:

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

where recall $L_\pi(\pi(i))$, defined in (4) is the latency of the node $\pi(i)$, which is the cumulative distance traveled on a tour before reaching $\pi(i)$. $\mathcal{F}_0$ is the true hypothesis space for the ML&TRP, so we will bound the complexity of this space, in terms of C_{budget}; this will show that the failure cost term may assist with generalization.

The proof idea of the main theorem below (Theorem 1) is to enlarge this class of functions just enough so that a bound on the covering number of $\mathcal{F}_0$ can be calculated. In the proof, we will construct two classes, $\mathcal{F}_1$ and $\mathcal{F}_2$ that are slightly larger than $\mathcal{F}_0$, but

smaller than $\mathcal{F}$ when C_{budget} is small enough. Then we will use a volumetric argument to bound the covering number of $\mathcal{F}_2$, which uses the volumes of spherical caps; the idea is to show that the value of C_{budget} affects the volume of the hypothesis space, and thus the covering number. We will show how the space of functions $\mathcal{F}_0$ is related to $\mathcal{F}_1$ and $\mathcal{F}_2$ in a precise manner later. We will defer defining $\mathcal{F}_1$ till then. For now, we define set $\mathcal{F}_2$ parametrized by a vector $a_{\text{budget}} \in \mathbb{R}^d$ as follows:

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

$\mathcal{F}_2$ is the intersection of the ball $\mathcal{F}$ with the halfspace defined by a_{budget}; it is a ball that is missing a spherical cap. The vector a_{budget} will capture the effect of C_{budget} in such a way that $\mathcal{F}_0 \subset \mathcal{F}_2$, which we will show within the proof of the Theorem 1. $\mathcal{F}_2$ is the space whose complexity we will bound, again within the proof of Theorem 1.

We will now define the vector a_{budget} in terms of C_{budget}. To do so, we first define d_i to be the shortest distance from the starting node (node 1) to node i for $i = 2, .., M$. We then define d_1 to be the length of the shortest tour that visits all the nodes and returns to node 1. This means $d_i \leq L_\pi(i)$ for $i = 1, ..., M$, and this inequality can become equality if the physical graph can be embedded into 1-dimensional Euclidean space (on a line). The vector a_{budget} is defined elementwise by:

$$a^j_{\text{budget}} = \frac{1}{C_{\text{budget}} - a_0} \left(\frac{e^{M_1 M_2}}{(1 + e^{M_1 M_2})^2} \right) \left(\sum_i d_i \tilde{x}^j_i \right) \text{ for } j = 1, .., d \tag{23}$$

where

$$a_0 = \left(M_1 M_2 \, \frac{e^{M_1 M_2}}{(1 + e^{M_1 M_2})^2} + \frac{1}{1 + e^{M_1 M_2}} \right) \sum_i d_i.$$

The main result is as follows:

Theorem 1 ***(Main Result)*** *Let $\mathcal{X} = \{x \in \mathbb{R}^d : \|x\|_2 \leq M_2\}$, $\mathcal{Y} = \{-1, 1\}$. Let $\mathcal{F}_0$ be defined as above with respect to $\{\tilde{x}_i\}_{i=1}^M$, $\tilde{x}_i \in \mathcal{X}$ (not necessarily random) and a corresponding physical graph. Let $\{x_i, y_i\}_{i=1}^m$ be a sequence of m instances drawn independently according to an unknown distribution $\mu_{\mathcal{X}\times\mathcal{Y}}$. Let M_{bound} be equal to $|e^{M_1 M_2} - e^{-M_1 M_2}|$. Then for any $\epsilon > 0$,*

$$\begin{aligned} & P\Big(\exists f \in \mathcal{F}_0 : |R^{\text{emp}}(f_\lambda, \{x_i, y_i\}_1^m) - R^{\text{true}}(f_\lambda)| > \epsilon\Big) \\ & \leq \quad 4\alpha(d, a_{\text{budget}}(C_{\text{budget}})) \left(\frac{32 M_1 M_2}{\epsilon} + 1 \right)^d \exp\left(\frac{-m\epsilon^2}{128 M^2_{bound}} \right), \end{aligned}$$

where

$$\alpha(d, a_{\text{budget}}(C_{\text{budget}})) := \frac{1}{2} + \frac{\|a_{\text{budget}}\|_2^{-1} + \frac{\epsilon}{32 M_2}}{M_1 + \frac{\epsilon}{32 M_2}} \frac{\Gamma\left[1 + \frac{d}{2}\right]}{\sqrt{\pi}\Gamma\left[\frac{d+1}{2}\right]} {}_2F_1\left(\tfrac{1}{2}, \tfrac{1-d}{2}; \tfrac{3}{2}; \left(\frac{\|a_{\text{budget}}\|_2^{-1} + \frac{\epsilon}{32 M_2}}{M_1 + \frac{\epsilon}{32 M_2}} \right)^2 \right) \tag{24}$$

or equivalently

$$\alpha(d, a_{\text{budget}}(C_{\text{budget}})) := 1 - \frac{1}{2} I_{1 - \left(\|a_{\text{budget}}\|_2^{-1} + \frac{\epsilon}{32 M_2}\right)^2 / \left(M_1 + \frac{\epsilon}{32 M_2}\right)^2} \left(\frac{d+1}{2}, \frac{1}{2} \right) \tag{25}$$

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

The term $\alpha(d, a_{\text{budget}}(C_{\text{budget}}))$ comes directly from formulae for the volume of spherical caps. Our goal is to establish that generalization can depend on C_{budget}. The value of C_{budget} enters into the bound through vector a_{budget}. As C_{budget} decreases, the norm $\|a_{\text{budget}}\|_2$ increases, and thus $\|a_{\text{budget}}\|_2^{-1}$ decreases, (25) and (24) decrease, and the whole bound decreases. This is the mechanism by which decreasing C_{budget} may improve generalization ability.

We will provide several lemmas leading to the proof of the theorem, but first we proceed to define the set $\mathcal{F}_1$, which is larger than $\mathcal{F}_0$ and smaller than $\mathcal{F}_2$. $\mathcal{F}_1$ is defined using a lower bound on the latencies $L_\pi(i)$, namely the minimum distances d_i. We have, for any collection of values $p(\tilde{x}_i) \geq 0$:

$$\sum_i d_i p(\tilde{x}_i) \leq \sum_i L_\pi(i) p(\tilde{x}_i) \leq C_{\text{budget}}.$$

This means that the class of functions which obey the constraint $\sum_i d_i p(\tilde{x}_i) \leq C_{\text{budget}}$ is larger than the class obeying $\sum_i L_\pi(i) p(\tilde{x}_i) \leq C_{\text{budget}}$. That is, $\mathcal{F}_0 \subseteq \mathcal{F}_1$ where

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

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

We provide some common definitions.

Definition 2 *Let $A \subseteq X$ be an arbitrary set and $(X, dist)$ 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 $dist(x, u) \leq \epsilon$.*
- *A is totally bounded if A has a finite ϵ-cover for all $\epsilon > 0$. The covering number of A is then defined as $N(\epsilon, A, dist) := \inf_{U \in \mathcal{U}} |U|$ where $\mathcal{U}$ is the set of all ϵ-covers for A.*
- *A set $R \subseteq X$ is ϵ-separated if $\forall x, y \in R, dist(x, y) > \epsilon$. The packing number $M(\epsilon, A, dist) := \sup_{R \in \mathcal{R}} |R|$, where $\mathcal{R}$ is the set of all ϵ-separated subsets of A.*

We have already seen the distribution $\mu_{\mathcal{X} \times \mathcal{Y}}$ before. For a general support set $\mathcal{B}$, we use $\mu_{\mathcal{B}}$ to represent a probability measure on it. We set (uppercase) B to be a random variable taking values in $\mathcal{B}$ according to $\mu_{\mathcal{B}}$, and (lowercase) b to be a realization of B. Further, we use the notation $\mu_{\mathcal{B}}^m$ to represent the empirical measure based on sample set $\{b_i\}_{i=1}^m = \{b_1, ..., b_m\}$. We use $L_2(\mu_{\mathcal{B}})$ to denote a space of functions defined on set $\mathcal{B}$ with the metric $\|f - g\|_{L_2(\mu_{\mathcal{B}})} = \int (f(b) - g(b))^2 d\mu_{\mathcal{B}}$. For our purposes, $\mathcal{B}$ will be either $\mathcal{X} \times \mathcal{Y}$ which we have already seen, or it will be the input space $\mathcal{X}$.

We will derive an upper bound for the covering number of $\mathcal{F}_1$ (and as a consequence $\mathcal{F}_0$) by finding a covering number bound for the more tractable $\mathcal{F}_2$. We will show that the vector a_{budget} is defined in such a way that $\mathcal{F}_2$ is a larger class than $\mathcal{F}_1$.

Lemma 3 *($\mathcal{F}_0$ **is contained in** $\mathcal{F}_2$)*

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

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

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

We now show $\mathcal{F}_1 \subseteq \mathcal{F}_2$. We first find two real numbers m_1 and m_0 so that they are the slope and intercept of a linear lower bound for the $p(\tilde{x}_i)$'s.

$$m_1 f_\lambda(\tilde{x}_i) + m_0 \leq p(\tilde{x}_i) = \frac{1}{1+e^{-f_\lambda(\tilde{x}_i)}}. \tag{26}$$

We know $f_\lambda : \mathcal{X} \to [-M_1M_2, M_1M_2]$ by the Cauchy-Schwarz inequality since $\forall f_\lambda \in \mathcal{F}, \forall x \in \mathcal{X}, |f_\lambda(x)| \leq M_1M_2$. Within the range $[-M_1M_2, M_1M_2]$ we lower bound the function $g(z) = 1/(1+e^{-z})$ by the line with slope

$$m_1 := g'(-M_1M_2) = \frac{e^{M_1M_2}}{(1+e^{M_1M_2})^2}$$

that intersects the point $(-M_1M_2, g(-M_1M_2))$, and thus has y-intercept

$$m_0 := M_1M_2 \frac{e^{M_1M_2}}{(1+e^{M_1M_2})^2} + \frac{1}{1+e^{M_1M_2}}.$$

In this way, (26) holds. These definitions and (26) further lead to the definition of a_{budget} as we show now:

$$\sum_i d_i p(\tilde{x}_i) \geq \sum_i d_i (m_1(\lambda \cdot \tilde{x}_i) + m_0) = m_1 \left(\sum_i d_i \tilde{x}_i \right) \cdot \lambda + m_0 \sum_i d_i = \tilde{a} \cdot \lambda + a_0 \tag{27}$$

where the intermediate d-dimensional vector $\tilde{a}$ is given elementwise as

$$\tilde{a}^j := m_1 \left(\sum_i d_i \tilde{x}_i^j \right) = \frac{e^{M_1M_2}}{(1+e^{M_1M_2})^2} \left(\sum_i d_i \tilde{x}_i^j \right) \text{ for } j = 1, ..., d \tag{28}$$

and,

$$a_0 = m_0 \sum_i d_i = \left(M_1M_2 \, \frac{e^{M_1M_2}}{(1+e^{M_1M_2})^2} + \frac{1}{1+e^{M_1M_2}} \right) \sum_i d_i.$$

Thus, we have that for $\forall \lambda \in \mathcal{F}_1$:

$$\tilde{a} \cdot \lambda + a_0 \leq \sum_{i=1}^{M} d_i p(\tilde{x}_i) \leq C_{\text{budget}}$$

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

This allows us to define a_{budget} using (28) as

$$a^j_{\text{budget}} = \frac{1}{C_{\text{budget}} - a_0} \left(\frac{e^{M_1M_2}}{(1+e^{M_1M_2})^2} \right) \left(\sum_i d_i \tilde{x}_i^j \right) \text{ for } j = 1, .., d$$

which is the same as (23). This vector is such that the set $\mathcal{F}_2$ is larger than $\mathcal{F}_1$.

■

Another way to obtain a suitable a_{budget} such that $\mathcal{F}_2$ is a tight superset of $\mathcal{F}_1$ is to minimize the distance of a hyperplane from the origin which is farther than the highest point on the surface of the non-linear constraint of $\mathcal{F}_1$. This can be achieved by solving a semi-infinite program (29):

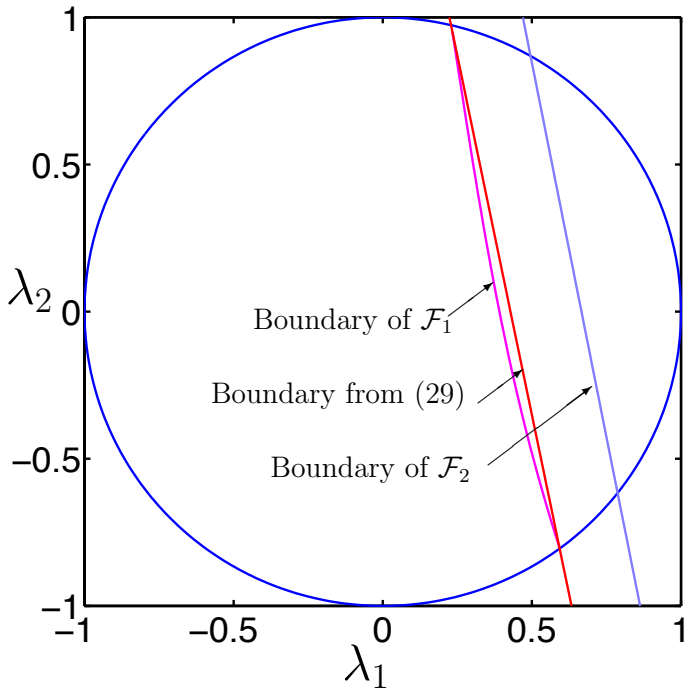


Figure 14: Hyperplanes upper bounding the nonlinear constraint in the description of $\mathcal{F}_1$. Here, $\lambda \in \mathbb{R}^2$. The ℓ_2 ball represents $\mathcal{F}$. For convenience, we have assumed $M_1 = 1$.

$$\max_{a} \|a\|_2^2$$
$$\text{s.t. } \forall \lambda \in \Lambda \cup \Lambda^0, a \cdot \lambda \leq 1$$
$$\text{where } \Lambda = \left\{ \lambda : \|\lambda\|_2 \leq M_1, \sum_{i=1}^{M} d_i \frac{1}{1+\exp\left(-\lambda \cdot \tilde{x}_i\right)} = C_{\text{budget}} \right\} \tag{29}$$
$$\text{and } \Lambda^0 = \left\{ \lambda : \|\lambda\|_2 = M_1, \sum_{i=1}^{M} d_i \frac{1}{1+\exp\left(-\lambda \cdot \tilde{x}_i\right)} \leq C_{\text{budget}} \right\}.$$

One can approximate the two sets Λ and Λ^0 in the program formulation by discretizing, and the semi-infinite program becomes a non-linear program. Figure (14) provides a 2-dimensional illustration of $\mathcal{F}, \mathcal{F}_1, \mathcal{F}_2$ and the approximate solution of the semi-infinite program. The semi-infinite program yields a tighter bound on $\mathcal{F}_1$ but is more expensive to compute.

Let $B_{M_1} = \{\lambda : \|\lambda\|_2 \leq M_1\}$ be a closed ball of radius M_1 in $\mathbb{R}^d$. Set B_{M_1} is used for constructing functions in $\mathcal{F}$. Let the half space corresponding to $\mathcal{F}_2$ be defined as

$$H_{\|a_{\text{budget}}\|_2^{-1}} := \{\lambda : a_{\text{budget}} \cdot \lambda \leq 1\}.$$

The value $\|a_{\text{budget}}\|_2^{-1}$ will be used in the volumetric argument. Because of rotational symmetry of B_{M_1}, the volume cut off by a hyperplane $a_{\text{budget}} \cdot \lambda = 1$ from B_{M_1} is determined only by its distance from the origin, which is $1/\|a_{\text{budget}}\|_2$. Such a portion (or its complement, if smaller) of a ball obtained from slicing the ball with a hyperplane is called a spherical cap. It can be parameterized by the distance of its (hyper)plane base from the center of the ball. The theorem below relates covering numbers of $\mathcal{F}$ and $\mathcal{F}_2$ in function space (considered in Lemma 3) to covering numbers of B_{M_1} and $B_{M_1} \cap H_{\|a_{\text{budget}}\|_2^{-1}}$ in $\mathbb{R}^d$.

Lemma 4 ***(Relating covering numbers in*** $\|\cdot\|_{L_2(\mu_{\mathcal{X}}^m)}$ ***to*** $\|\cdot\|_2$***)***

a. $\sup_{\mu_{\mathcal{X}}^m} N(\epsilon, \mathcal{F}, \|\cdot\|_{L_2(\mu_{\mathcal{X}}^m)}) \leq N(\epsilon/M_2, B_{M_1}, \|\cdot\|_2)$

b. $\sup_{\mu_{\mathcal{X}}^m} N(\epsilon, \mathcal{F}_2, \|\cdot\|_{L_2(\mu_{\mathcal{X}}^m)}) \leq N(\epsilon/M_2, B_{M_1} \cap H_{\|a_{\text{budget}}\|_2^{-1}}, \|\cdot\|_2)$.

Proof Each element $f \in \mathcal{F}$ corresponds to at least one element of B_{M_1} by definition of $\mathcal{F}$. Choose any distribution $\mu_{\mathcal{X}}^m$. Consider two elements $\lambda_f, \lambda_g \in B_{M_1}$ corresponding to functions $f, g \in \mathcal{F} \subset L_2(\mu_{\mathcal{X}}^m)$. Then,

$$\begin{aligned}
\|f-g\|_{L_2(\mu_{\mathcal{X}}^m)}^2 &= \frac{1}{m}\sum_{i=1}^{m}(f(x_i)-g(x_i))^2 \\
&= \frac{1}{m}\sum_{i=1}^{m}\left((\lambda_f-\lambda_g)\cdot x_i\right)^2 \\
&\leq \frac{1}{m}\sum_{i=1}^{m}\|\lambda_f-\lambda_g\|_2^2\|x_i\|_2^2 \text{ (Cauchy-Schwarz to each term)} \\
&\leq \|\lambda_f-\lambda_g\|_2^2\left(\frac{1}{m}\sum_{i=1}^{m}M_2^2\right) \text{ (since } \sup_{x\in\mathcal{X}}\|x\|_2 \leq M_2) \\
&= \|\lambda_f-\lambda_g\|_2^2 M_2^2.
\end{aligned}$$

Consider a minimal ϵ/M_2-cover $\{\lambda_r\}_r$ for B_{M_1} where λ_r corresponds to a function $r \in \mathcal{F}$. Then by definition, $\forall \lambda \in B_{M_1}, \exists \lambda_r : \|\lambda - \lambda_r\|_2 \leq \epsilon/M_2$. Thus, picking any two such elements λ_f, λ_g in a ball of radius ϵ/M_2 around λ_r, we see that, the corresponding functions f, g belong to a ball of radius ϵ measured using distance in $L_2(\mu_{\mathcal{X}}^m)$ by the inequality above. The centers of these ϵ-balls in $L_2(\mu_{\mathcal{X}}^m)$ form an ϵ-cover for $\mathcal{F}$. The size of this set is equal to $N(\epsilon/M_2, B_{M_1}, \|\cdot\|_2)$ (which is the size of ϵ/M_2-cover for B_{M_1}). The size of the minimal ϵ-cover of $\mathcal{F}$ will be less than or equal to this size. Hence, $N(\epsilon, \mathcal{F}, \|\cdot\|_{L_2(\mu_{\mathcal{X}}^m)}) \leq N(\epsilon/M_2, B_{M_1}, \|\cdot\|_2)$. Taking a supremum over all $\mu_{\mathcal{X}}^m$, we obtain the first inequality of the lemma. The same argument also works for the second inequality. ■

Now that we have bounded the covering numbers of $\mathcal{F}$ and $\mathcal{F}_2$ by covering numbers of sets in d-dimensional λ-space using Lemma 4, we will proceed to use the notion of volumes in $\mathbb{R}^d$ to bound the latter quantities. In doing so, we will need the volume of a spherical cap in $\mathbb{R}^d$. For notation, let the volume of a set $A \subset \mathbb{R}^d$ be represented as $Vol(A)$. For example, $Vol(B_1) = \frac{\pi^{d/2}}{\Gamma[d/2+1]}$.

Lemma 5 *(**Volume of spherical caps**) Let the volume of ball B_{M_1} in $\mathbb{R}^d$ be denoted as $Vol(B_{M_1})$. Given a d-dimensional vector a, let $z = \|a\|_2^{-1}$ be a number and $H_z = \{\lambda : a \cdot \lambda \leq 1\}$ be a half space parameterized by z. Let the spherical cap be denoted by $B_{M_1} \cap H_z'$ where the cap is at a distance z (measured from the base of the cap to the center of the ball), and H_z' represents the complement half space ($H_z \cup H_z' = \mathbb{R}^d$). Then,*

$$Vol(B_{M_1} \cap H_z') \quad = Vol(B_{M_1}) \left(\frac{1}{2} - \frac{z}{M_1} \frac{\Gamma\left[1+\frac{d}{2}\right]}{\sqrt{\pi}\Gamma\left[\frac{d+1}{2}\right]} {}_2F_1\left(\frac{1}{2}, \frac{1-d}{2}; \frac{3}{2}; \left(\frac{z}{M_1}\right)^2 \right) \right)$$

where ${}_2F_1(a, b; c; d)$ is the hypergeometric function. Alternatively,

$$Vol(B_{M_1} \cap H_z') \quad = Vol(B_{M_1}) \frac{1}{2} I_{1-z^2/M_1^2}\left(\frac{d+1}{2}, \frac{1}{2}\right)$$

where $I_x(e, f)$ is the regularized incomplete beta function.

Proof See Li (2011) and references therein. ■

Note that for $0 \leq z \leq M_1, Vol(B_{M_1} \cap H_z') \leq Vol(B_{M_1})$. If $Vol(B_{M_1} \cap H_z') \leq \frac{1}{2} Vol(B_{M_1})$, then the volume of the spherical cap reduces rapidly with increasing dimension d. This has an important effect on the covering number as a function of dimension d, and ultimately on the generalization bound too. Figure 15 illustrates this point by showing the volume on one side of the hyperplane as the hyperplane moves through the ball, for various values of the dimension d. If d is fairly large, then the volume decreases dramatically as the hyperplane passes through the center of the ball.

There is a well-known relationship between packing numbers and covering numbers which we will make use of in proving Theorem 7 below.

Lemma 6 *(**Packing and covering numbers**) For every (pseudo) metric space $(X, dist)$, $A \subseteq X$, and $\epsilon > 0$,*

$$N(\epsilon, A, dist) \leq M(\epsilon, A, dist).$$

Proof See Theorem 4 in Kolmogorov and Tikhomirov (1959) or Theorem 12.1 in Anthony and Bartlett (1999) for a proof of this classical result. ■

We now use the expression for volume of the spherical cap in Lemma 5 and the relation between packing and covering of Lemma 6 in obtaining bounds for the covering numbers of

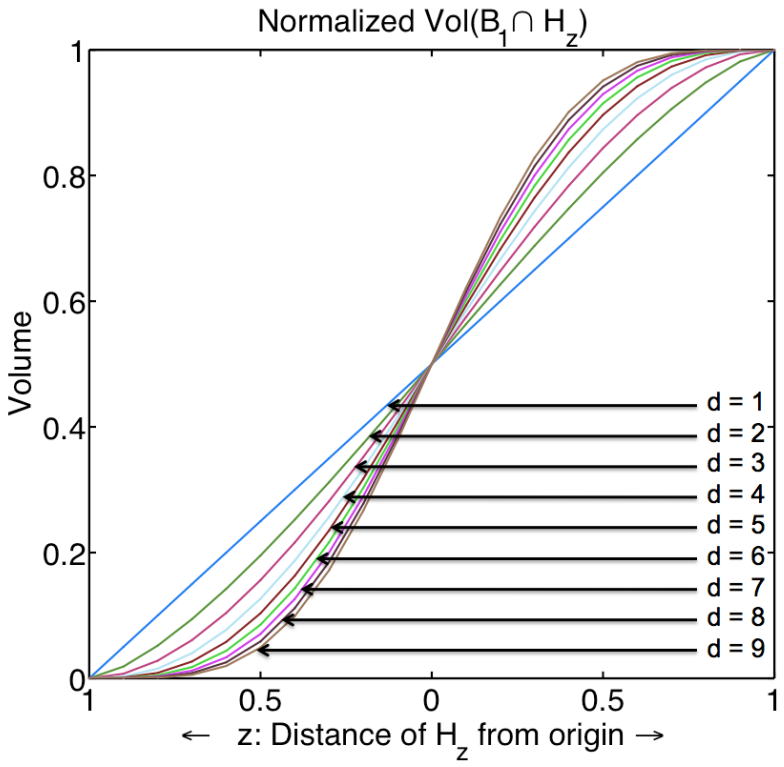


Figure 15: $Vol(B_1 \cap H_z)$ vs. z, in other words, volume of a unit ℓ_2-ball intersected with a halfspace, as a function of the distance of the hyperplane from the center of the ball. This also illustrates the dependence of $\alpha(d, a_{\text{budget}}(C_{\text{budget}}))$ on C_{budget}. To the left of 0 on x-axis, the distance increases from the origin but volume decreases, eventually reaching a value of 0. To the right of 0 on x-axis, the distance increases from the origin again, but now the volume increases, eventually reaching the volume of the ball or value 1 (normalized).

subsets of $\mathbb{R}^d$ which appeared in Lemma 4. Note that the relationship between a spherical cap and its complement is: $Vol\left(B_{M_1} \cap H'_z\right) = Vol(B_{M_1}) - Vol\left(B_{M_1} \cap H_z\right)$.

Theorem 7 ***(Bound on Covering Numbers)***

$$\begin{aligned} N(\epsilon/M_2, B_{M_1}, \|\cdot\|_2) &\leq \left(\frac{2M_1M_2}{\epsilon}+1\right)^d \\ N\left(\epsilon/M_2, B_{M_1} \cap H_{\|a\|_2^{-1}} \|\cdot\|_2\right) &\leq \left(\frac{Vol\left(B_{M_1+\frac{\epsilon}{2M_2}} \cap H_{\|a\|_2^{-1}+\frac{\epsilon}{2M_2}}\right)}{Vol\left(B_{M_1+\frac{\epsilon}{2M_2}}\right)}\right)\left(\frac{2M_1M_2}{\epsilon}+1\right)^d. \end{aligned}$$

Proof Both statements involve a volumetric argument. There are various versions of proof for the first part. For example, see Section 3 of Kolmogorov and Tikhomirov (1959), Lemma 4.10 in Pisier (1989), Lorentz (1966) and Lemma 3 in Cucker and Smale (2002) among others. We will provide an argument along these lines. Let $\lambda_1, .., \lambda_M$ be an optimal ϵ-packing for B_{M_1}. That is, $M = M(\epsilon, B_{M_1}, \|\cdot\|_2)$. The volume of $B_{M_1+\epsilon/2}$ (an extra $\epsilon/2$ added so that the packing elements can lie within the boundary) is,

$$Vol(B_{M_1+\epsilon/2}) = Vol(B_1)(M_1+\epsilon/2)^d$$

where $Vol(B_1)$ is the volume of a unit ball in dimension d. The volume of an $\epsilon/2$ ball with packing element λ_i as the center is:

$$Vol(B_{\epsilon/2}+\lambda_i) = Vol(B_{\epsilon/2}) = Vol(B_1)(\epsilon/2)^d.$$

Since the sum of the volume of the $\epsilon/2$ balls should be less than or equal to the volume of the extended ball $B_{M_1+\epsilon/2}$ (else one of the packing elements λ_i will be outside the boundary of B_{M_1} contradicting the definition of packing) we have:

$$M(\epsilon, B_{M_1}, \|\cdot\|_2)Vol(B_1)(\epsilon/2)^d \leq Vol(B_1)(M_1+\epsilon/2)^d.$$

Scaling ϵ to ϵ/M_2 and using the inequality between minimal covering and maximal packing numbers from Lemma 6 we obtain the first stated result.

To show the second part, let the volume of the complement of the spherical cap be $Vol(B_{M_1} \cap H_{\|a\|_2^{-1}})$; we need to find an upper bound for the minimal ϵ/M_2-cover of this set. We can do that by scaling a minimal ϵ-cover, which we find now. By extending the boundary of $B_{M_1} \cap H_{\|a\|_2^{-1}}$ by $\epsilon/2$ we can bound the maximal packing number $M(\epsilon, B_{M_1} \cap H_{\|a\|_2^{-1}}, \|\cdot\|_2)$ as follows.

$$M(\epsilon, B_{M_1} \cap H_{\|a\|_2^{-1}}, \|\cdot\|_2)Vol(B_1)(\epsilon/2)^d \leq Vol(B_{M_1+\epsilon/2} \cap H_{\|a\|_2^{-1}+\epsilon/2})$$

$$\begin{aligned} M(\epsilon, B_{M_1} \cap H_{\|a\|_2^{-1}}, \|\cdot\|_2) &\leq \left(\frac{Vol\left(B_{M_1+\epsilon/2} \cap H_{\|a\|_2^{-1}+\epsilon/2}\right)}{Vol(B_1)}\right)\frac{1}{(\epsilon/2)^d} \\ &= \left(\frac{Vol\left(B_{M_1+\epsilon/2} \cap H_{\|a\|_2^{-1}+\epsilon/2}\right)}{Vol(B_1)}\right)\frac{1}{(\epsilon/2)^d}\frac{(M_1+\epsilon/2)^d}{(M_1+\epsilon/2)^d} \\ &= \left(\frac{Vol\left(B_{M_1+\epsilon/2} \cap H_{\|a\|_2^{-1}+\epsilon/2}\right)}{Vol(B_{M_1+\epsilon/2})}\right)\frac{(M_1+\epsilon/2)^d}{(\epsilon/2)^d}. \end{aligned}$$

Again, scaling ϵ to ϵ/M_2 and using the relationship between $N(\epsilon, A, dist)$ and $M(\epsilon, A, dist)$ in Lemma 6 yields the second result. ■

Thus we have so far shown the relationship between covering numbers of $\mathcal{F}_0$, $\mathcal{F}_1$, and $\mathcal{F}_2$ in terms of a certain metric in Lemma 3, we have shown how those covering numbers are related to covering numbers in $\ell_2(\mathbb{R}^d)$ in Lemma 4, we have shown how the latter covering numbers relate to volumes in $\ell_2(\mathbb{R}^d)$ in Theorem 7, and we have shown how to compute one of these volumes in Lemma 5.

To complete the proof of Theorem 1, we will use a relation between the covering number of a class of loss functions of some set $\mathcal{G}$ and the covering number of the set $\mathcal{G}$ itself. We will also use a uniform convergence bound of Pollard (1984). We state the latter below:

Theorem 8 *(**Pollard 1984**) Let $l_{\mathcal{G}}$ be a set of functions on $\mathcal{X} \times \mathcal{Y}$ with $0 \leq l(f_\lambda(x), y) \leq M_{bound}$, $\forall l \in l_{\mathcal{G}}$ and $\forall (x, y) \in \mathcal{X} \times \mathcal{Y}$. Let $\{x_i, y_i\}_1^m$ be a sequence of m instances drawn independently according to $\mu_{\mathcal{X}\times\mathcal{Y}}$. Then for any $\epsilon > 0$,*

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

Proof See Theorem 24 in Pollard (1984) (also in Zhang, 2002, Theorem 1). Note that the constants have been refined in other works since the first result and we have left the original constants intact here. ■

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

Lemma 9 *(**Relating $l_{\mathcal{G}}$ to $\mathcal{G}$**) If every function from function class $l_{\mathcal{G}}$ represented as $l : f(\mathcal{X}) \times \mathcal{Y} \mapsto \mathbb{R}, f \in \mathcal{G}$, is Lipschitz in its first argument with Lipschitz constant $\mathcal{L}$, then the covering number of $l_{\mathcal{G}}$ is related to the covering number of $\mathcal{G}$ by*

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

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

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

This implies, given $\{x_i, y_i\}_{i=1}^m$, if $\hat{\mathcal{G}}$ is a minimal $\epsilon/\mathcal{L}$-cover of $\mathcal{G}$ in $L_1(\mu^m_{\mathcal{X}})$, we can construct an ϵ-cover of $l_{\mathcal{G}}$ in $L_1(\mu^m_{\mathcal{X}\times\mathcal{Y}})$ as

$$\hat{l}_{\mathcal{G}} = \{l_{f_i} : f_i \in \hat{\mathcal{G}}.\}$$

The size of the minimal ϵ-cover will be smaller than the size of such an ϵ-cover. Taking supremum over all empirical distributions, we get the desired result. ■

The logistic loss $\log(1+e^{-yf(x)})$ when viewed as a function of $f(x)$ has a Lipschitz constant $\mathcal{L} \leq 1$. For a similar result using the squared loss see Lemma 17.4 of Anthony and Bartlett (1999).

Theorem 8 and Lemma 9 involve L_1 covering numbers, but our covering number bounds start with an L_2 metric in Lemma 4. So we need to switch from L_1 to L_2 metric. The following lemma uses the identity $\|f-g\|_{L_1(\mu_X^m)} \leq \|f-g\|_{L_2(\mu_X^m)}$ (true because of Jensen's inequality applied to norms) to relate the two.

Lemma 10 $N(\epsilon, A, \|\cdot\|_{L_1(\mu_{\mathcal{X}}^m)}) \leq N(\epsilon, A, \|\cdot\|_{L_2(\mu_{\mathcal{X}}^m)})$.

Proof See for a version, Lemma 10.5 in Anthony and Bartlett (1999). ■

Finally, we can prove the main result.

Proof *(Of Theorem 1)* Starting from the expectation term on the right hand side of Theorem 8 using $\mathcal{F}_0$ as $\mathcal{G}$,

$$
\begin{aligned}
& E[N(\epsilon/16, l_{\mathcal{F}_0}, \|\cdot\|_{L_1(\mu^m_{\mathcal{X}\times\mathcal{Y}})})] \\
& \leq \sup_{\mu^m_{\mathcal{X}\times\mathcal{Y}}} N(\epsilon/16, l_{\mathcal{F}_0}, \|\cdot\|_{L_1(\mu^m_{\mathcal{X}\times\mathcal{Y}})}) \text{ bounding expectation by supremum} \\
& \leq \sup_{\mu^m_{\mathcal{X}}} N\left(\frac{\epsilon}{16\mathcal{L}}, \mathcal{F}_0, \|\cdot\|_{L_1(\mu^m_{\mathcal{X}})}\right) \text{ from Lemma 9} \\
& \leq \sup_{\mu^m_{\mathcal{X}}} N\left(\frac{\epsilon}{16\mathcal{L}}, \mathcal{F}_0, \|\cdot\|_{L_2(\mu^m_{\mathcal{X}})}\right) \text{ from Lemma 10} \\
& \leq \sup_{\mu^m_{\mathcal{X}}} N\left(\frac{\epsilon}{16\mathcal{L}}, \mathcal{F}_2, \|\cdot\|_{L_2(\mu^m_{\mathcal{X}})}\right) \text{ from Lemma 3} \\
& \leq N\left(\frac{\epsilon}{16\cdot 1\cdot M_2}, B_{M_1} \cap H_{\|a_{\text{budget}}\|_2^{-1}}, \|\cdot\|_2\right) \text{ from Lemma 4 and substituting } \mathcal{L}=1 \\
& \leq \left(\frac{Vol\left(B_{M_1+\frac{\epsilon}{32M_2}} \cap H_{\|a_{\text{budget}}\|_2^{-1}+\frac{\epsilon}{32M_2}}\right)}{Vol(B_{M_1+\frac{\epsilon}{32M_2}})}\right)\left(\frac{32M_1M_2}{\epsilon}+1\right)^d \text{ from Theorem 7} \\
& = \alpha(d, a_{\text{budget}}(C_{\text{budget}}))\left(\frac{32M_1M_2}{\epsilon}+1\right)^d \text{ from Lemma 5.}
\end{aligned}
$$

The above step uses the relationship between the spherical cap and its complement along with Lemma 5,

$$Vol\left(B_{M_1} \cap H'_{\|a_{\text{budget}}\|_2^{-1}}\right) = Vol(B_{M_1}) - Vol\left(B_{M_1} \cap H_{\|a_{\text{budget}}\|_2^{-1}}\right).$$

Using this bound on $E[N(\epsilon/16, l_{\mathcal{F}_0}, \|\cdot\|_{L_1(\mu^m_{\mathcal{X}\times\mathcal{Y}})})]$ within Theorem 8 gives the result. ■

6. Discussion and Related Works

One main focus of this work was to discuss a tradeoff between learning error and operational cost. In doing so, we showed a new way in which data dependent regularization can influence an algorithm's prediction ability, formalized through generalization bounds. There is a vast literature on regularization, but in the past it has been used to impose prior beliefs (e.g., "structure" such as sparsity like Tibshirani, 1996, shrinking certain coefficients towards each

other), robustness (e.g., to obtain a large "margin" which is the distance from the decision boundary to the nearest training instance like Vapnik, 1998), or additional distributional information (semi-supervised learning, see for instance Chapelle et al., 2006). Out of these, only semi-supervised learning uses unlabeled data, but our problem differs in that our unlabeled data does not need to be drawn from the same distribution as the training data, and thus does not necessarily provide any distributional information. It provides instead information about the practical cost of following the algorithm's recommendations.

The ML&TRP relates to literature on both machine learning and optimization (time-dependent traveling salesman problems). In machine learning, our work bears a slight resemblance to work on graph-based regularization (Agarwal, 2006; Belkin et al., 2006; Zhou et al., 2004), but their goal is to obtain probability estimates that are smoothed on a graph with suitably designed edge weights. On the other hand, our goal is to obtain, in addition to probability estimates, a low-cost route for traversing a very different graph with edge weights that are physical distances. Our work contributes to the literature on the TRP and related problems by adding the new dimension of probabilistic estimation at the nodes. We create new adaptations of modern techniques (Fischetti et al., 1993; Eijl van, 1995; Lechmann, 2009) within our work for solving the TRP part of the ML&TRP.

In addition to the above, we developed cost models that apply to routing problems. For the power grid application and other maintenance applications, $\{d_{ij}\}_{ij}$ in (4) correspond to physical distances. It is possible to use the techniques developed here for more abstract routing problems, for instance, network scheduling or network routing problems, where distance on the graph does not necessarily correspond to a physical distance. There are other works that schedule events based on a linearly increasing cost model (see for instance Anily et al., 1998).

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

If we were able to find an efficient method for approximately solving the TRP subproblem, it could allow us to compute solutions to the ML&TRP significantly faster. Constant factor approximation algorithms for the standard (unweighted) TRP have been developed in several works (Goemans and Kleinberg, 1998; Blum et al., 1994; Arora and Karakostas, 2006; Archer et al., 2008; Archer and Blasiak, 2010). These schemes typically have (quasi-) polynomial time guarantees and approximate up to a constant ratio of the optimal standard TRP objective value. The constant factors, however, are at least 3.59. Heuristic methods could also be used for solving the standard TRP and related problems (Dewilde et al., 2010; Salehipour et al., 2010) that can potentially be adapted to solve the weighted TRP.

There are some difficulties in doing this because the heuristics depend on the exact way the cost is defined. For example, Dewilde et al. (2010) solve a variation of the TRP which cannot easily be adapted for solving the weighted TRP problem. Lechmann (2009) has a survey of the various applications and solution techniques of the different versions of the TRP problem.

There could be many variations on the setup for the ML&TRP. In some applications, real time sensor measurements are available, and it is possible to automatically turn off the equipment when it fails in order to prevent more failures from occurring. This is not possible for the power grid application, since it is not possible (and not desirable) to turn off the electricity supply in the secondary electrical distribution network, but it may be possible in other applications.

7. Conclusion

In this work, we present a machine learning algorithm that takes into account the way its recommendations will be ultimately used. This algorithm takes advantage of uncertainty in the model in order to potentially find a much more practical solution. Including these operating costs is a new way of incorporating "structure" into machine learning algorithms, and we plan to explore this in other ways in ongoing work. We discussed the potential tradeoff between generalization and operating cost for the specific application to the ML&TRP. In doing so, we showed a new way in which data dependent regularization can influence an algorithm's prediction ability, formalized through generalization bounds.

Acknowledgements

We gratefully acknowledge support from an International Fulbright Science and Technology Award, the MIT Energy Initiative, and the National Science Foundation under Grant No IIS-1053407. Thanks to Shai Ben-David for helpful discussions.

References

Shivani Agarwal. Ranking on graph data. In *Proceedings of the 23rd International Conference on Machine Learning*, 2006.

Shoshana Anily, Celia A. Glass, and Refael Hassin. The scheduling of maintenance service. *Discrete Applied Mathematics*, 82(1-3):27–42, 1998.

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

Aaron Archer and Anna Blasiak. Improved approximation algorithms for the minimum latency problem via prize-collecting strolls. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 429–447, 2010.

Aaron Archer, Asaf Levin, and David P. Williamson. A faster, better approximation algorithm for the minimum latency problem. *SIAM J. Comput.*, 37(5):1472–1498, 2008.

Sanjeev Arora and George Karakostas. A $2 + \epsilon$ approximation algorithm for the k-MST problem. *Math. Program.*, 107(3):491–504, 2006.

Fran Barbera, Helmut Schneider, and Peter Kelle. A condition based maintenance model with exponential failures and fixed inspection intervals. *The Journal of the Operational Research Society*, 47(8):pp. 1037–1045, 1996.

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

Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

Avrim Blum, Prasad Chalasani, Don Coppersmith, Bill Pulleyblank, Prabhakar Raghavan, and Madhu Sudan. On the minimum latency problem. *ArXiv Mathematics e-prints*, September 1994.

Pierre Bonami, Lorenz T. Biegler, Andrew R. Conn, Gérard Cornuéjols, Ignacio E. Grossmann, Carl D. Laird, Jon Lee, Andrea Lodi, François Margot, Nicolas W. Sawaya, and Andreas Wächter. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5(2):186–204, 2008.

Olivier Bousquet. New approaches to statistical learning theory. *Annals of the Institute of Statistical Mathematics*, 55(2):371–389, 2003.

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

Imre Csiszár and G. Tusnády. Information geometry and alternating minimization procedures. *Statistics and Decisions*, 1(Suppl.):205–237, 1984.

Felipe Cucker and Steve Smale. On the mathematical foundations of learning. *Bulletin-American Mathematical Society*, 39(1):1–50, 2002.

Thijs Dewilde, Dirk Cattrysse, Sofie Coene, Frits C. R. Spieksma, and Pieter Vansteenwegen. Heuristics for the Traveling Repairman Problem with Profits. *10th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems*, pages 34–44, 2010.

C. A. Eijl van. A polyhedral approach to the delivery man problem. Technical report, Memorandum COSOR 95–19, Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands, 1995.

Matteo Fischetti, Gilbert Laporte, and Silvano Martello. The delivery man problem and cumulative matroids. *Oper. Res.*, 41:1055–1064, November 1993.

Michel Goemans and Jon Kleinberg. An improved approximation ratio for the minimum latency problem. *Mathematical Programming*, 82:111–124, 1998.

David Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and computation*, 100(1):78–150, 1992.

Aiwina Heng, Andy C.C. Tan, Joseph Mathew, Neil Montgomery, Dragan Banjevic, and Andrew K.S. Jardine. Intelligent condition-based prediction of machinery reliability. *Mechanical Systems and Signal Processing*, 23(5):1600 – 1614, 2009.

Waltraud Huyer and Arnold Neumaier. Global optimization by multilevel coordinate search. *J. of Global Optimization*, 14:331–355, June 1999.

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.

Miriam Lechmann. The traveling repairman problem - an overview. *Diplomarbeit, Universitat Wein*, pages 1–79, 2009.

Shengqiao Li. Concise Formulas for the Area and Volume of a Hyperspherical Cap. *Asian Journal of Mathematics & Statistics*, 4(1):66–70, 2011.

George G. Lorentz. Metric entropy and approximation. *Bull. Am. Math. Soc.*, 72:903–937, 1966.

Marzio Marseguerra, Enrico Zio, and Luca Podofillini. Condition-based maintenance optimization by means of genetic algorithms and monte carlo simulation. *Reliability Engineering & System Safety*, 77(2):151 – 165, 2002.

Shahar Mendelson and Roman Vershynin. Entropy and the combinatorial dimension. *Inventiones Mathematicae*, 152(1):37–55, 2003.

Isabel Méndez-Díaz, Paula Zabala, and Abilio Lucena. A new formulation for the traveling deliveryman problem. *Discrete Applied Mathematics*, 156(17):3223–3237, 2008.

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

Jean-Claude Picard and Maurice Queyranne. The time-dependent traveling salesman problem and its application to the tardiness problem in one-machine scheduling. *Operations Research*, 26(1): 86–110, January–February 1978.

Gilles Pisier. *The volume of convex bodies and Banach space geometry*, volume 94. Cambridge University Press, Cambridge, 1989.

David Pollard. *Convergence of stochastic processes*. Springer, 1984.

Luis Miguel Rios. Algorithms for derivative-free optimization. *PhD thesis, University of Illinois at Urbana-Champaign*, pages 1–133, 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, David Waltz, Roger Anderson, Albert Boulanger, Ansaf Salleb-Aouissi, Maggie Chow, Haimonti Dutta, Phil Gross, Bert Huang, Steve Ierome, Delfina Isaac, Artie Kressner, Rebecca Passonneau, Axinia Radeva, and Leon Wu. Machine learning for the New York City power grid. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011. accepted subject to minor revision.

Amir Salehipour, Kenneth Sorensen, Peter Goos, and Olli Bräysy. Efficient GRASP+ VND and GRASP+ VNS metaheuristics for the traveling repairman problem. *4OR: A Quarterly Journal of Operations Research*, pages 1–21, 2010.

John Shawe-Taylor and Nello Cristianini. On the generalization of soft margin algorithms. *IEEE Transactions on Information Theory*, 48(10):2721–2735, 2002.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996. ISSN 0035-9246.

Theja Tulabandhula, Cynthia Rudin, and Patrick Jaillet. The machine learning and traveling repairman problem. In *Proceedings of the Second International Conference on Algorithmic Decision Theory*, 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.

Vladimir Naumovich Vapnik. *Statistical learning theory*, volume 2. Wiley New York, 1998.

Andrés Weintraub, J. Aboud, C. Fernandez, G. Laporte, and E. Ramirez. An emergency vehicle dispatching system for an electric utility in Chile. *Journal of the Operational Research Society*, pages 690–696, 1999.

Zhengguo Xu, Yindong Ji, and Donghua Zhou. A new real-time reliability prediction method for dynamic systems based on on-line fault prediction. *IEEE Transactions on Reliability*, 58(3): 523–538, 2009.

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

Dengyong Zhou, Jason Weston, Arthur Gretton, Olivier Bousquet, and Bernhard Schölkopf. Ranking on data manifolds. In *Advances in Neural Information Processing Systems 16*, pages 169–176. MIT Press, 2004.