Smart “Predict, then Optimize”

Adam N. Elmachtoub
Department of Industrial Engineering and Operations Research, Columbia University, New York, NY 10027,
adam@ieor.columbia.edu

Paul Grigas
Department of Industrial Engineering and Operations Research, University of California, Berkeley, CA 94720,
pgrigas@berkeley.edu

Many real-world analytics problems involve two significant challenges: prediction and optimization. Due to the typically complex nature of each challenge, the standard paradigm is predict-then-optimize. By and large, machine learning tools are intended to minimize prediction error and do not account for how the predictions will be used in the downstream optimization problem. In contrast, we propose a new and very general framework, called Smart “Predict, then Optimize” (SPO), which directly leverages the optimization problem structure, i.e., its objective and constraints, for designing better prediction models. A key component of our framework is the SPO loss function which measures the decision error induced by a prediction.

Training a prediction model with respect to the SPO loss is computationally challenging, and thus we derive, using duality theory, a convex surrogate loss function which we call the SPO+ loss. Most importantly, we prove that the SPO+ loss is statistically consistent with respect to the SPO loss under mild conditions. Our SPO+ loss function can tractably handle any polyhedral, convex, or even mixed-integer optimization problem with a linear objective. Numerical experiments on shortest path and portfolio optimization problems show that the SPO framework can lead to significant improvement under the predict-then-optimize paradigm, in particular when the prediction model being trained is misspecified.

Key words: prescriptive analytics; data-driven optimization; machine learning; linear regression

1. Introduction

In many real-world analytics applications of operations research, a combination of both machine learning and optimization are used to make decisions. Typically, the optimization model is used to generate decisions, while a machine learning tool is used to generate a prediction model that predicts key unknown parameters of the optimization model. Due to the inherent complexity of both tasks, a broad purpose approach that is often employed in analytics practice is the *predict-then-optimize* paradigm.

For example, consider a vehicle routing problem that may be solved several times a day. First, a previously trained prediction model provides predictions for the travel time on all edges of a road network based on current traffic, weather, holidays, time, etc. Then, an optimization solver provides near-optimal routes using the predicted travel times as input. We emphasize that most solution systems for real-world analytics problems involve some component of both prediction and optimization (see Angalakudati et al. (2014), Chan et al. (2012), Deo et al. (2015), Gallien et al. (2015), Cohen et al. (2017), Besbes et al. (2015), Mehrotra et al. (2011), Chan et al. (2013), Ferreira et al. (2015) for recent examples and recent expositions by Simchi-Levi (2013), den Hertog and Postek (2016), Deng et al. (2018), Mišić and Perakis (2019)). Except for a few limited options, machine learning tools do not effectively account for how the predictions will be used in a downstream optimization problem. In this paper, we provide a general framework called Smart "Predict, then Optimize" (SPO) for training prediction models that effectively utilize the structure of the nominal optimization problem, i.e., its constraints and objective. *Our SPO framework is fundamentally designed to generate prediction models that aim to minimize decision error, not prediction error.*

Our SPO approach fundamentally maintains the paradigm of sequentially predicting and then optimizing. However, when training our prediction model, the structure of the nominal optimization problem is explicitly used. The quality of a prediction is *not* measured based on prediction error such as least squares loss or other popular loss functions. Instead, in the SPO framework, the quality of a prediction is measured by the decision error. That is, suppose a prediction model is trained using historical feature data $(x_1, \ldots, x_n)$ and associated parameter data $(c_1, \ldots, c_n)$. Let $(\hat{c}_1, \ldots, \hat{c}_n)$ denote the predictions of the parameters under the trained model. The least squares loss, for example, measures error with the squared norm $\|c_i - \hat{c}_i\|_2^2$, completely ignoring the decisions induced by the predictions. In contrast, the *SPO loss* is the true cost of the decision induced by $\hat{c}_i$ minus the optimal cost under the true parameters c_i.

In this paper, we focus on predicting unknown parameters that appear linearly in the objective function, namely, the cost vector of any linear, convex, or integer optimization problem. The core of our SPO framework is a new loss function for training prediction models. Since this loss function is difficult to work with, significant effort revolves around deriving a surrogate loss function that is convex and therefore can be optimized efficiently. To show the validity of this surrogate, we prove a highly desirable statistical consistency property, and show it performs well empirically compared to standard predict-then-optimize approaches. Empirically, our framework works quite well, and suggests that even when the prediction task may be challenging, making near-optimal decisions is still possible. We believe our SPO framework provides a clear foundation for designing machine learning tools that can be leveraged in real-world optimization settings.

Our contributions may be summarized as follows:

1. We first formally define a new loss function, which we call the SPO loss, that measures the error in predicting the cost vector of a nominal optimization problem with linear, convex, or integer constraints. The loss corresponds to the suboptimality gap – with respect to the true/historical cost vector – due to implementing a possibly incorrect decision induced by the predicted cost vector. Unfortunately, the SPO loss function can be nonconvex and discontinuous in the predictions, implying that training ML models under the SPO loss may be challenging.
2. Given the intractability of the SPO loss function, we develop a surrogate loss function which we call the SPO+ loss. This surrogate loss function is derived using a sequence of upper bounds motivated by duality theory, a data scaling approximation, and a first-order approximation. The resulting SPO+ loss function is convex in the predictions. Moreover, when training a linear model to predict the objective coefficients of a linear program, only a linear optimization problem needs be solved to minimize the SPO+ loss. We provide a general algorithm for minimizing the SPO+ loss which is based on stochastic gradient descent.
3. We prove a fundamental connection to classical machine learning under a very simple and special instance of our SPO framework. Namely, under this instance the SPO loss is exactly the 0-1 classification loss and the SPO+ loss is exactly the hinge loss. The hinge loss is the basis of the popular SVM method and is a surrogate loss to approximately minimize the 0-1 loss, and thus our framework generalizes this concept to a very wide family of optimization problems with constraints.
4. We prove a key consistency result of the SPO+ loss function, which further motivates its use. Namely, under full distributional knowledge, minimizing the SPO+ loss function is in fact equivalent to minimizing the SPO loss if two mild conditions hold: the distribution of the cost vector is continuous and symmetric about its mean. For example, these assumptions are satisfied by a Gaussian random variable. *This consistency property is widely regarded as an essential property of any surrogate loss function across the statistics and machine learning literature.* For example, the famous hinge loss and logistic loss functions are consistent with the 0-1 classification loss.

5. Finally, we validate our framework through numerical experiments on the shortest path problem. We test our SPO framework against standard predict-then-optimize approaches, and evaluate the out of sample performance with respect to the SPO loss. Generally, the value of our SPO framework increases as the degree of model misspecification increases. This is precisely due to the fact the SPO framework makes "better" wrong predictions, essentially "tricking" the optimization problem into finding near-optimal solutions. Remarkably, a linear model trained using SPO+ even dominates a state-of-the-art random forests algorithm, even when the ground truth is highly nonlinear.

1.1. Applications

Settings where the input parameters (cost vectors) of an optimization problem need to be predicted from contextual (feature) data are numerous. Let us now highlight a few, of potentially many, application areas for the SPO framework.

Vehicle Routing. In numerous applications, the cost of each edge of a graph needs to be predicted before making a routing decision. The cost of an edge typically corresponds to the expected length of time a vehicle would need to traverse the corresponding edge. For clarity, let us focus on one important example – the shortest path problem. In the shortest path problem, one is given a weighted directed graph, along with an origin node and destination node, and the goal is to find a sequence of edges from the origin to the destination at minimum possible cost. A well-known fact is that the shortest path problem can be formulated as a linear optimization problem, but there are also alternative specialized algorithms such as the famous Dijkstra's algorithm (see, e.g., Ahuja et al. (1993)). The data used to predict the cost of the edges may incorporate the length, speed limit, weather, season, day, and real-time data from mobile applications such as Google Maps and Waze. Simply minimizing prediction error may not suffice nor be appropriate, as over- or under-predictions have starkly different effects across the network. The SPO framework would ensure that the predicted weights lead to shortest paths, and would naturally emphasize the estimation of edges that are critical to this decision. See Figure 1 in Section 2 for an in-depth example.

Inventory Management. In inventory planning problems such as the economic lot sizing problem (Wagner and Whitin (1958)) or the joint replenishment problem (Levi et al. (2006)), the demand is the key input into the optimization model. In practical settings, demand is highly nonstationary and can depend on historical and contextual data such as weather, seasonality, and competitor sales. The decisions of when to order inventory are captured by a linear or integer optimization model, depending on the complexity of the problem. Under a common formulation (see Levi et al. (2006), Cheung et al. (2016)), the demand appears linearly in the objective, which is convenient for the SPO framework. The goal is to design a prediction model that maps feature data to demand predictions, which in turn lead to good inventory plans.

Portfolio Optimization. In financial services applications, the returns of potential investments need to be somehow estimated from data, and can depend on many features which typically include historical returns, news, economic factors, social media, and others. In portfolio optimization, the goal is to find a portfolio with the highest return subject to a constraint on the total risk, or variance, of the portfolio. While the returns are often highly dependent on auxiliary feature information, the variances are typically much more stable and are not as difficult nor sensitive to predict. Our SPO framework would result in predictions that lead to high performance investments that satisfy the desired level of risk. A least squares loss approach places higher emphasis on estimating higher valued investments, even if the corresponding risk may not be ideal. In contrast, the SPO framework directly accounts for the risk of each investment when training the prediction model.

1.2. Related Literature

Perhaps the most related work is that of Kao et al. (2009), who also directly seek to train a machine learning model that minimizes loss with respect to a nominal optimization problem. In their framework, the nominal problem is an unconstrained quadratic optimization problem, where the unknown parameters appear in the linear portion of the objective. Their work does not extend to settings where the nominal optimization problem has constraints, which our framework does. Donti et al. (2017) proposes a heuristic to address a more general setting than that of Kao et al. (2009), and also focus on the case of quadratic optimization. These works also bypass issues of non-uniqueness of solutions of the nominal problem (since their problem is strongly convex), which must be addressed in our setting to avoid degenerate prediction models.

In Rudin and Vahn (2014), ML models are trained to directly predict the optimal solution of a newsvendor problem from data. Tractability and statistical properties of the method are shown as well as its effectiveness in practice. However, it is not clear how this approach can be used when there are constraints, since feasibility issues may arise.

The general approach in Bertsimas and Kallus (2014) considers the problem of accurately estimating an unknown optimization objective using machine learning models, specifically ML models where the predictions can be described as a weighted combination of training samples, e.g., nearest neighbors and decision trees. In their approach, they estimate the objective of an instance by applying the same weights generated by the ML model to the corresponding objective functions of those samples. This approach differs from standard predict-then-optimize *only* when the objective function is nonlinear in the unknown parameter. Note that the unknown parameters of all the applications mentioned in Section 1.1 appear linearly in the objective. Moreover, the training of the ML models does not rely on the structure of the nominal optimization problem, in contrast to the SPO framework.

The approach in Tulabandhula and Rudin (2013) relies on minimizing a loss function that combines the prediction error with the operational cost of the model on an unlabeled dataset. However, the operational cost is with respect to the predicted parameters, and not the true parameters. We also note that our SPO loss, while mathematically different, is similar in spirit to the notion of relative regret introduced in Lim et al. (2012) in the specific context of portfolio optimization with historical return data and without features. Other approaches for finding near-optimal solutions from data include operational statistics (Liyanage and Shanthikumar (2005), Chu et al. (2008)), sample average approximation (Kleywegt et al. (2002), Schütz et al. (2009), Bertsimas et al. (2014)), and robust optimization (Bertsimas and Thiele (2006), Bertsimas et al. (2013), Wang et al. (2016)). There has also been some recent progress on submodular optimization from samples (Balkanski et al. (2016, 2017)). These approaches typically do not have a clear way of using feature data, nor do they directly consider how to train a machine learning model to predict optimization parameters.

Another related stream of work is in data-driven inverse optimization, where feasible or optimal solutions to an optimization problem are observed and the objective function has to be learned (Aswani et al. (2015), Keshavarz et al. (2011), Chan et al. (2014), Bertsimas et al. (2015), Esfahani et al. (2015)). In these problems, there is typically a single unknown objective, and no previous samples of the objective are provided. We also note there have been recent approaches for regularization (Ban et al. (2016)) and model selection (Den Boer and Sierag (2016), Sen and Deng (2017)) in the context of an optimization problem.

Finally, we note that our framework is related to the general setting of structured prediction (see, e.g., Osokin et al. (2017), Goh and Jaillet (2016) and the references therein). Motivated by problems in computer vision and natural language processing, structured prediction is a version of multiclass classification that is concerned with predicting structured objects, such as sequences or graphs, from feature data. The SPO framework presents a new paradigm for structured prediction where the structured objects are decision variables associated with a natural optimization problem.

2. "Predict, then Optimize" Framework

We now describe the "Predict, then Optimize" framework which is central to many applications of optimization in practice. Specifically, we assume that there is a nominal optimization problem of interest with a linear objective where the decision variable $w \in \mathbb{R}^d$ and feasible region $S \subseteq \mathbb{R}^d$ are well-defined and known with certainty. However, the cost vector of the objective, $c \in \mathbb{R}^d$, is not observed directly, and rather an associated feature vector $x \in \mathbb{R}^p$ is observed. Let $\mathcal{D}_x$ be the conditional distribution of c given x. The goal for the decision maker is to solve, for any new instance characterized by x,

$$\min_{w \in S} \mathbb{E}_{c \sim \mathcal{D}_x}[c^\top w | x] \ = \ \min_{w \in S} \mathbb{E}_{c \sim \mathcal{D}_x}[c | x]^\top w \ . \tag{1}$$

The predict-then-optimize framework relies on using a prediction for $\mathbb{E}_{c \sim \mathcal{D}_x}[c|x]$, which we denote by $\hat{c}$, and solving the deterministic version of the optimization problem based on $\hat{c}$, i.e., $\min_{w \in S} \hat{c}^\top w$. Our primary interests in this paper concern defining suitable loss functions for the predict-then-optimize framework, examining their properties, and developing algorithms for training prediction models using these loss functions.

We now formally list the key ingredients of our framework:

1. *Nominal optimization problem*, which is of the form

$$\begin{aligned} P(c): \quad z^*(c) := \ & \min_{w} \ c^T w \\ & \text{s.t.} \ \ w \in S \ , \end{aligned} \tag{2}$$

where $w \in \mathbb{R}^d$ are the decision variables, $c \in \mathbb{R}^d$ is the problem data describing the linear objective function, and $S \subseteq \mathbb{R}^d$ is a nonempty, compact (i.e., closed and bounded), and convex set representing the feasible region. Since S is assumed to be fixed and known with certainty, every problem instance can be described by the corresponding cost vector, hence the dependence on c in (2). When solving a particular instance where c is unknown, a prediction for c is used instead. We assume access to a practically efficient optimization oracle, $w^*(c)$, that returns a solution of $P(c)$ for any input cost vector. For instance, if

(2) corresponds to a linear, conic, or even a combinatorial or mixed-integer optimization problem (in which case S can be implicitly described as a convex set), then a commercial optimization solver or a specialized algorithm suffices for $w^*(c)$.

2. *Training data* of the form $(x_1, c_1), (x_2, c_2), \ldots, (x_n, c_n)$, where $x_i \in \mathcal{X}$ is a feature vector representing contextual information associated with c_i.
3. A *hypothesis class* $\mathcal{H}$ of cost vector prediction models $f : \mathcal{X} \to \mathbb{R}^d$, where $\hat{c} := f(x)$ is interpreted as the predicted cost vector associated with feature vector x.
4. A *loss function* $\ell(\cdot,\cdot) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$, whereby $\ell(\hat{c}, c)$ quantifies the error in making prediction $\hat{c}$ when the realized (true) cost vector is actually c.

Given the loss function $\ell(\cdot,\cdot)$ and the training data $(x_1, c_1), \ldots, (x_n, c_n)$, the empirical risk minimization principle states that we should determine a prediction model $f^* \in \mathcal{H}$ by solving the optimization problem

$$\min_{f \in \mathcal{H}} \ \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), c_i) \ . \tag{3}$$

Provided with the prediction model f^*, a natural decision rule is induced for the nominal optimization problem when presented with a feature vector x, namely the optimal solution with respect to the predicted cost vector, $w^*(f^*(x))$, is chosen. Example 1 contextualizes our framework in the context of a network optimization problem.

Example 1 (Network Flow). An example of the nominal optimization problem is a minimum cost network flow problem, where the decisions are how much flow to send on each edge of the network. We assume that the underlying graph is provided to us, e.g., the road network of a city. The feasible region S represents flow conservation, capacity, and required flow constraints on the underlying network. The cost vector c is not known with certainty, but can be estimated from data x which can include features of time, day, edge lengths, most recent observed cost, and so on. An example of the hypothesis class is the set of linear prediction models given by $\mathcal{H} = \{f : f(x) = Bx \text{ for some } B \in \mathbb{R}^{d \times p}\}$. The linear model can be trained, for example, according to the mean squared error loss function, i.e., $\ell(\hat{c}, c) = \frac{1}{2}\|\hat{c} - c\|_2^2$. The corresponding empirical risk minimization problem to find the best linear model B^* then becomes $\min_B \ \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2}\|Bx_i - c_i\|_2^2$. The decision rule to find the optimal network flow given a feature vector x is $w^*(B^*x)$. □

In standard applications of the "Predict, then Optimize" framework, as in Example 1, the loss function that is used is completely independent of the nominal optimization problem.

In other words, the underlying structure of the optimization problem $P(\cdot)$ does not factor into the loss function and therefore the training of the prediction model. For example, when $\ell(\hat{c}, c) = \frac{1}{2}\|\hat{c} - c\|_2^2$, this corresponds to the mean squared error. Moreover, if $\mathcal{H}$ is a set of linear predictors, then (3) reduces to a standard least squares linear regression problem. In contrast, our focus in Section 3 is on the construction of loss functions that intelligently measure errors in predicting cost vectors and leverage problem structure when training the prediction model.

Useful Notation. Let p be the dimension of a feature vector, d be the dimension of a decision vector, and n be the number of training samples. Let $W^*(c) := \arg\min_{w \in S} \{c^T w\}$ denote the set of optimal solutions of $P(\cdot)$, and let $w^*(\cdot) : \mathbb{R}^d \to S$ denote a particular *oracle* for solving $P(\cdot)$. That is, $w^*(\cdot)$ is a fixed deterministic mapping such that $w^*(c) \in W^*(c)$. Note that nothing special is assumed about the mapping $w^*(\cdot)$, hence $w^*(c)$ may be regarded as an arbitrary element of $W^*(c)$. Let $\xi_S(\cdot) : \mathbb{R}^d \to \mathbb{R}$ denote the support function of S, which is defined by $\xi_S(c) := \max_{w \in S}\{c^T w\}$. Since S is compact, $\xi_S(\cdot)$ is finite everywhere, the maximum in the definition is attained for every $c \in \mathbb{R}^d$, and note that $\xi_S(c) = -z^*(-c) = c^T w^*(-c)$ for all $c \in \mathbb{R}^d$. Recall also that $\xi_S(\cdot)$ is a convex function. For a given convex function $h(\cdot) : \mathbb{R}^d \to \mathbb{R}$, recall that $g \in \mathbb{R}^d$ is a subgradient of $h(\cdot)$ at $c \in \mathbb{R}^d$ if $h(c') \geq h(c) + g^T(c' - c)$ for all $c' \in \mathbb{R}^d$, and the set of subgradients of $h(\cdot)$ at c is denoted by $\partial h(c)$. For two matrices $B_1, B_2 \in \mathbb{R}^{d \times p}$, the trace inner product is denoted by $B_1 \bullet B_2 := \mathrm{trace}(B_1^T B_2)$.

3. SPO Loss Functions

Herein, we introduce several loss functions that fall into the predict-then-optimize paradigm, but that are also *smart* in that they take the nominal optimization problem $P(\cdot)$ into account when measuring errors in predictions. We refer to these loss functions as Smart "Predict, then Optimize" (SPO) loss functions. As a starting point, let us consider a true SPO loss function that exactly measures the excess cost incurred when making a suboptimal decision due to an imprecise cost vector prediction. Following the PO paradigm, given a cost vector prediction $\hat{c}$, a decision $w^*(\hat{c})$ is implemented based on solving $P(\hat{c})$. After the decision $w^*(\hat{c})$ is implemented, the cost incurred is with respect to the cost vector c that is *actually realized*. The excess cost due to the fact that $w^*(\hat{c})$ may be suboptimal with respect to c is then $c^T w^*(\hat{c}) - z^*(c)$. Definition 1 formalizes this true SPO loss associated with making the prediction $\hat{c}$ when the actual cost vector is c, given a particular oracle $w^*(\cdot)$ for $P(\cdot)$.

DEFINITION 1. Given a cost vector prediction $\hat{c}$ and a realized cost vector c, the *true SPO loss* $\ell_{\mathrm{SPO}}^{w^*}(\hat{c}, c)$ w.r.t. optimization oracle $w^*(\cdot)$ is defined as $\ell_{\mathrm{SPO}}^{w^*}(\hat{c}, c) := \; c^T w^*(\hat{c}) - z^*(c)$.

Note that there is an unfortunate deficiency in Definition 1, which is the dependence on the particular oracle $w^*(\cdot)$ used to solve (2). Practically speaking, this deficiency is not a major issue since we should usually expect $w^*(\hat{c})$ to be a unique optimal solution, i.e., we should expect $W^*(\hat{c})$ to be a singleton. (Note that if any solution from $W^*(\hat{c})$ may be used by the loss function, then the loss function essentially becomes $\min_{w \in W^*(\hat{c})} c^T w - z^*(c)$. Thus, a prediction model would then be incentivized to always predict $\hat{c} = 0$ since $W^*(0) = S$, and therefore the loss would be trivially 0.)

In any case, if one wishes to address the dependence on the particular oracle $w^*(\cdot)$ in Definition 1, then it is most natural to "break ties" by presuming that the implemented decision has worst-case behavior with respect to c. Definition 2 is an alternative SPO loss function that does not depend on the particular choice of the optimization oracle $w^*(\cdot)$.

DEFINITION 2. Given a cost vector prediction $\hat{c}$ and a realized cost vector c, the (unambiguous) *true SPO loss* $\ell_{\mathrm{SPO}}(\hat{c}, c)$ is defined as $\ell_{\mathrm{SPO}}(\hat{c}, c) := \max_{w \in W^*(\hat{c})} \left\{ c^T w \right\} - z^*(c)$.

Note that Definition 2 presents a version of the true SPO loss that upper bounds the version from Definition 1, i.e., it holds that $\ell_{\mathrm{SPO}}^{w^*}(\hat{c}, c) \leq \ell_{\mathrm{SPO}}(\hat{c}, c)$ for all $\hat{c}, c \in \mathbb{R}^d$. As mentioned previously, the distinction between Definitions 1 and 2 is only relevant in degenerate cases. In the results and discussion herein, we work with the unambiguous true SPO loss given by Definition 2. Related results may often be inferred for the version of the true SPO loss given by Definition 1 by recalling that Definition 2 upper bounds Definition 1 and that the two loss functions are almost always equal except for degenerate cases where $W^*(\hat{c})$ has multiple optimal solutions.

Notice that $\ell_{\mathrm{SPO}}(\hat{c}, c)$ is impervious to the scaling of $\hat{c}$, in other words it holds that $\ell_{\mathrm{SPO}}(\alpha \hat{c}, c) = \ell_{\mathrm{SPO}}(\hat{c}, c)$ for all $\alpha > 0$. This property is intuitive since the true loss associated with prediction $\hat{c}$ should only depend on the optimal *solution* of $P(\cdot)$, which does not depend on the scaling of $\hat{c}$. Moreover, this property is also shared by the 0-1 loss function in binary classification problems. Namely, labels can take values of ± 1 and the prediction model predicts values in $\mathbb{R}$. If the predicted value has the same sign as the true value, the loss is 0, and otherwise the loss is 1. Therefore, the 0-1 loss function is also independent of the scale on the predictions. This similarity is not a coincidence; in fact, Proposition 1 illustrates that binary classification is a special case of the SPO framework.

Proposition 1. *The 0-1 loss function associated with binary classification is a special case of the SPO loss function.*

Proof. Let $d = 1$ and the feasible region be the interval $S = [-1/2, +1/2]$. Here the "cost vector" c corresponds to a binary class label, i.e., c can take one of two possible values, -1 or $+1$. (However, the predicted cost vector is allowed to be any real number.) Notice that, for both possible values of c, it holds that $z^*(c) = -1/2$. There are three cases to consider for the prediction $\hat{c}$: *(i)* if $\hat{c} > 0$ then $W^*(\hat{c}) = \{-1/2\}$ and $\ell_{\text{SPO}}(\hat{c}, c) = (1 - c)/2$, *(ii)* if $\hat{c} < 0$ then $W^*(c) = \{+1/2\}$ and $\ell_{\text{SPO}}(\hat{c}, c) = (1 + c)/2$, and *(iii)* if $\hat{c} = 0$ then $W^*(\hat{c}) = S = [-1/2, +1/2]$ and $\ell_{\text{SPO}}(\hat{c}, c) = (1 + |c|)/2 = 1$. Thus, we have $\ell_{\text{SPO}}(\hat{c}, c) = 0$ when $\hat{c}$ and c share the same sign, and $\ell_{\text{SPO}}(\hat{c}, c) = 1$ otherwise. Therefore, ℓ_{SPO} is exactly the 0-1 loss function. □

Now, given the training data, we are interested in determining a cost vector prediction model with minimal true SPO loss. Therefore, given the previous definition of the true SPO loss $\ell_{\text{SPO}}(\cdot, \cdot)$, the prediction model would be determined by following the empirical risk minimization principle as in (3), which leads to the following optimization problem:

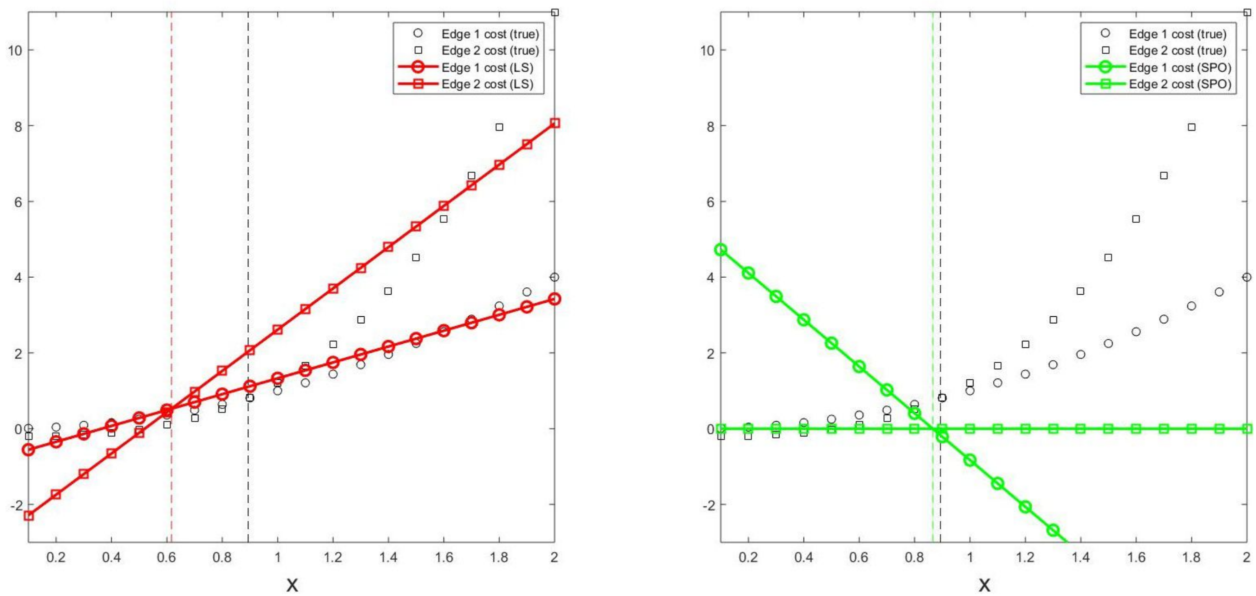
$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ell_{\text{SPO}}(f(x_i), c_i) \ . \tag{4}$$

Unfortunately, the above optimization problem is difficult to solve, both in theory and in practice. Indeed, for a fixed c, $\ell_{\text{SPO}}(\cdot, c)$ may not even be continuous in $\hat{c}$ since $w^*(\hat{c})$ (and the entire set $W^*(\hat{c})$) may not be continuous in $\hat{c}$. Moreover, since Proposition 1 demonstrates that our framework captures binary classification, solving (4) is at least as difficult as optimizing the 0-1 loss function, which may be NP-hard in many cases (Ben-David et al. 2003). We are therefore motivated to develop approaches for producing "reasonable" approximate solutions to (4) that *(i)* outperform standard PO approaches, and *(ii)* are applicable to large-scale problems where the number of training samples n and/or the dimension of the hypothesis class $\mathcal{H}$ may be very large.

3.1. An Illustrative Example

In order to build intuition, we now compare the SPO loss against the classical least squares (LS) loss function via an illustrative example. Consider a very simple shortest path problem with two nodes s and t. There are two edge that go from s to t, edge 1 and edge 2. Thus, a cost vector c is 2-dimensional in this setting, and the goal is to simply choose the edge with the lower cost. We shall not observe c directly at the decision-making time, but rather just a 1-dimensional feature x associated with the vector c. Our data consists of 20 (x_i, c_i) pairs where x_i are between 0 and 2, and c_i are generated nonlinearly as a function of x_i. See Figure 1 for the raw data represented by the black points. The dotted black line represents the optimal decision boundary – to the right, one should always select edge 1 and to the left, one should always select edge 2.

Figure 1 Illustrative example.



Note. The circles correspond to edge 1 costs and the squares correspond to edge 2 costs. Red lines and points correspond to the least squares fit and predictions, while green lines and points correspond to the SPO fit and predictions. The vertical dotted lines correspond to the decision boundaries under the true and prediction models

The goal of the decision maker is to predict the cost of each edge from the feature using a simple linear regression model. The intersection of the two lines (corresponding to each edge) will signal the decision boundary in the predict-then-optimize framework. The decision maker shall try both the SPO and LS loss functions to do the linear regression. In the left panel of Figure 1, we plot the best LS fit to the data, and in the right panel we plot the best SPO fit. (In fact, the right side is also simultaneously an example of the best SPO+ fit which we derive in Section 3.2.)

One can see from Figure 1 that the LS lines very closely approximate the nonlinear data, although the decision boundary for LS is quite far from the optimal decision boundary. For any value of x between the dotted black and red lines, the decision maker will choose the wrong edge. In contrast, the SPO lines do not remotely approximate the data, yet its decision boundary is nearly-optimal. In fact, the SPO lines have 0 training error, despite not fitting the data at all. The key intuitition is that the SPO loss guides the choice of lines into making optimal decisions, and does not concern itself with predictions. Of course, a convex combination of SPO and LS loss may be used to overcome the unusual looking lines generated. In fact, there are many optimal solutions to the ERM problem for the SPO loss, all of which just require that the intersection of the lines occurs between the x values of 0.8 and 0.9.

3.2. The SPO+ Loss Function

In this section, we focus on deriving a tractable surrogate loss function that reasonably approximates $\ell_{\text{SPO}}(\cdot,\cdot)$. In fact, our surrogate function $\ell_{\text{SPO+}}(\cdot,\cdot)$, which we call the SPO+ loss function, can be derived from a series of three upper bounds on $\ell_{\text{SPO}}(\cdot,\cdot)$. We shall first derive the SPO+ loss function, and then in Section 3.3 motivate why each upper bound is a reasonable approximation. Ideally, when finding the prediction model that minimizes the empirical risk using the SPO+ loss, this prediction model will also approximately minimize (4), the empirical risk using the SPO loss.

To begin the derivation of the SPO+ loss, we first observe that for any $\alpha \in \mathbb{R}$, the SPO loss can be written as

$$\ell_{\text{SPO}}(\hat{c}, c) = \max_{w \in W^*(\hat{c})} \left\{c^T w - \alpha \hat{c}^T w\right\} + \alpha z^*(\hat{c}) - z^*(c) \tag{5}$$

since $z^*(\hat{c}) = \hat{c}^T w$ for all $w \in W^*(\hat{c})$. Clearly, replacing the constraint $w \in W^*(\hat{c})$ with $w \in S$ in (5) results in an upper bound. Since this is true for all values of α, then the first upper bound is

$$\ell_{\text{SPO}}(\hat{c}, c) \ \leq \ \min_{\alpha} \left\{ \max_{w \in S} \left\{c^T w - \alpha \hat{c}^T w\right\} + \alpha z^*(\hat{c}) \right\} - z^*(c) \ . \tag{6}$$

Now simply setting $\alpha = 2$ in the right-hand side of (6) yields an even greater upper bound:

$$\ell_{\text{SPO}}(\hat{c}, c) \ \leq \ \max_{w \in S} \left\{c^T w - 2\hat{c}^T w\right\} + 2z^*(\hat{c}) - z^*(c) \ . \tag{7}$$

Finally, since $w^*(c)$ is a feasible solution of $P(\hat{c})$, an even greater upper bound is

$$\ell_{\text{SPO}}(\hat{c}, c) \ \leq \ \max_{w \in S} \left\{c^T w - 2\hat{c}^T w\right\} + 2\hat{c}^T w^*(c) - z^*(c) \ . \tag{8}$$

The final upper bound (8) is exactly what we refer to as the SPO+ loss function, which we formally state in Definition 3. (Recall that $\xi_S(\cdot)$ is the support function of S, i.e., $\xi_S(c) := \max_{w \in S}\{c^T w\}$.)

DEFINITION 3. Given a cost vector prediction $\hat{c}$ and a realized cost vector c, the *surrogate SPO+ loss* is defined as $\ell_{\mathrm{SPO+}}(\hat{c},c) := \xi_S(c-2\hat{c}) + 2\hat{c}^T w^*(c) - z^*(c)$.

Next, we state the following proposition which formally shows that, in addition to the SPO+ loss being an upper bound on the SPO loss, the SPO+ loss function is convex in $\hat{c}$. (This follows immediately from the convexity of the support function $\xi_S(\cdot)$.) Note that while the SPO+ loss is convex in $\hat{c}$, in general it is not differentiable since $\xi_S(\cdot)$ is not generally differentiable. However, we can easily compute subgradients of the SPO+ loss by utilizing the oracle $w^*(\cdot)$, namely since $w^*(-\tilde{c}) \in \partial\xi_S(\tilde{c})$ the chain rule implies that $2(w^*(c) - w^*(2\hat{c}-c)) \in \partial\ell_{\mathrm{SPO+}}(\hat{c},c)$. We exploit this fact in developing computational approaches in Section 5.

PROPOSITION 2. *Given a fixed realized cost vector c, it holds that $\ell_{\mathrm{SPO+}}(\hat{c},c)$ is a convex function of the cost vector prediction $\hat{c}$. Moreover, $\ell_{\mathrm{SPO}}(\hat{c},c) \leq \ell_{\mathrm{SPO+}}(\hat{c},c)$ for all $\hat{c} \in \mathbb{R}^d$.*

The convexity of this SPO+ loss function is also shared by the hinge loss function, which is a convex upper bound for the 0-1 loss function. Recall that the hinge loss given a prediction $\hat{c}$ is $\max\{0, 1-\hat{c}\}$ if the true label is 1 and $\max\{0, 1+\hat{c}\}$ if the true label is -1. More concisely, the hinge loss can be written as $\max\{0, 1-c\hat{c}\}$ where $c \in \{-1,+1\}$ is the true label. The hinge loss is central to the support vector machine (SVM) method, where it is used as a convex surrogate to minimize 0-1 loss. In fact, in the same setting as Proposition 1 where the SPO loss captures the 0-1 loss, Proposition 3 shows that the SPO+ loss function exactly matches the hinge loss. Moreover, the hinge loss satisfies a key consistency property with respect to the 0-1 loss (Steinwart (2002)), which justifies its use in practice. In Section 4 we show a similar consistency result for the SPO+ loss with respect to the SPO loss under some mild conditions.

PROPOSITION 3. *The hinge loss is equivalent to the SPO+ loss, under the same conditions where the 0-1 loss is equivalent to the SPO loss.*

Proof. In the same setup as Proposition 1, we have that $S = [-1/2, +1/2]$ and $c \in \{-1,+1\}$ corresponds to the true label. Note that $\xi_S(c-\hat{c}) = \frac{1}{2}|c-\hat{c}|$ and $w^*(c) = -c/2$ for $c \in \{-1,+1\}$. Therefore,

$$\ell_{\mathrm{SPO+}}(\hat{c},c) = \tfrac{1}{2}|c-\hat{c}| - \hat{c}c/2 + 1/2 = \tfrac{1}{2}|1-\hat{c}c| - \hat{c}c/2 + 1/2 = \max\{0, 1-\hat{c}c\} \ ,$$

where the second equality follows since $c \in \{-1,+1\}$. Thus, in this setting, $\ell_{\mathrm{SPO+}}$ is precisely the hinge loss. $\square$

Applying the ERM principle as in (4) to the SPO+ loss yields the following optimization problem for selecting the prediction model:

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ell_{\text{SPO+}}(f(x_i), c_i) \ . \tag{9}$$

Much of the remainder of the paper describes results concerning problem (9). In Section 4 we demonstrate the aforementioned Fisher consistency result, in Section 5 we describe several computational approaches for solving problem (9), and in Section 6 we demonstrate that (9) often offers superior practical performance over standard PO approaches. Next, we provide a theoretically motivated justification for using the SPO+ loss.

3.3. Justifying the SPO+ Loss Function

In the following, we provide intuitive and theoretical justification for each of the upper bounds, (6)-(8), that were used in the derivation of the surrogate SPO+ loss function. Our reasoning is confirmed by the computational results in Section 6, which evaluate the performance of the SPO+ loss function in various problem instances.

Justification of upper bound (6). Since the first upper bound in (6) involves optimizing over α, in a sense one may view (6) as a "weak duality" bound. It turns out that this intuition can be made precise by applying Lagrangian duality within the definition of the SPO loss function, and moreover strong duality actually holds in this case. Thus there is no loss in approximation from this step. Proposition 4 formalizes this result.

PROPOSITION 4. *For any cost vector prediction $\hat{c} \in \mathbb{R}^d$ and realized cost vector $c \in \mathbb{R}^d$, the true SPO loss function $\ell_{\text{SPO}}(\hat{c}, c) := \max_{w \in W^*(\hat{c})} \{c^T w\} - z^*(c)$ may be expressed as*

$$\ell_{\text{SPO}}(\hat{c}, c) \ = \ \inf_{\alpha} \{\xi_S(c - \alpha\hat{c}) + \alpha z^*(\hat{c})\} - z^*(c) \ . \tag{10}$$

Proof. We will actually prove that

$$\ell_{\text{SPO}}(\hat{c}, c) \ = \ \inf_{\alpha \geq 0} \{\xi_S(c - \alpha\hat{c}) + \alpha z^*(\hat{c})\} - z^*(c) \ , \tag{11}$$

where notice that the inf in (11) is over $\alpha \geq 0$ as opposed to $\alpha \in \mathbb{R}$ in (10). Then, given (6), it is clear that (11) also implies that (10) holds. The proof of (11) employs Lagrangian duality (see, e.g., Bertsekas (1999) and the references therein). First, note that the set of optimal solutions with respect to $\hat{c}$, $W^*(\hat{c}) := \arg\min_{w\in S}\left\{\hat{c}^T w\right\}$ may be expressed as $W^*(\hat{c}) = S \cap \{w \in \mathbb{R}^d : \hat{c}^T w \leq z^*(\hat{c})\}$. Therefore, it holds that:

$$\begin{array}{lcl} \max_w \ c^T w & = & \max_w \ c^T w \\ \text{s.t.} \ \ w \in W^*(\hat{c}) & & \text{s.t.} \ w \in S \\ & & \quad \hat{c}^T w \leq z^*(\hat{c}) \ . \end{array} \tag{12}$$

Let us introduce a scalar Lagrange multiplier $\alpha \in \mathbb{R}_+$ associated with the inequality constraint "$\hat{c}^T w \leq z^*(\hat{c})$" on the right side of (12) and then form the Lagrangian:

$$L(w,\alpha) := c^T w + \alpha(z^*(\hat{c}) - \hat{c}^T w) \ .$$

The dual function $q(\cdot) : \mathbb{R}_+ \to \mathbb{R}$ is defined in the standard way and satisfies:

$$\begin{aligned} q(\alpha) := \max_{w\in S} \ L(w,\alpha) &= \max_{w\in S}\left\{c^T w + \alpha(z^*(\hat{c}) - \hat{c}^T w)\right\} \\ &= \max_{w\in S}\left\{(c-\alpha\hat{c})^T w\right\} + \alpha z^*(\hat{c}) \\ &= \xi_S(c-\alpha\hat{c}) + \alpha z^*(\hat{c}) \ . \end{aligned}$$

Weak duality then implies that $\max_{w\in W^*(\hat{c})}\left\{c^T w\right\} \ \leq \ \inf_{\alpha\geq 0} q(\alpha)$ and hence:

$$\ell_{\text{SPO}}(\hat{c},c) \ = \ \max_{w\in W^*(\hat{c})}\left\{c^T w\right\} - z^*(c) \ \leq \ \inf_{\alpha\geq 0}\left\{\xi_S(c-\alpha\hat{c}) + \alpha z^*(\hat{c})\right\} - z^*(c) \ .$$

To prove (11), we demonstrate that strong duality holds by applying Theorem 4.3.8 of Borwein and Lewis (2010). In our setting, the primal problem is the problem on the right-hand side of (12). This problem corresponds to the primal minimization problem in Borwein and Lewis (2010) by considering the objective function given by $-c^T w + I_S(w)$ and the constraint function $\hat{c}^T w - z^*(\hat{c})$. (Note that $I_S(w)$ is the convex indicator function equal to 0 when $w \in S$ and $+\infty$ otherwise.) Since S is a compact and convex set, we satisfy all of the assumptions of Theorem 4.3.8 of Borwein and Lewis (2010) and hence strong duality holds. □

Justification of upper bound (7). The justification for the upper bound in (7) can be seen by first reformulating the true SPO loss ERM problem (4) using the alternate characterization of the SPO loss from Proposition 4. The SPO loss ERM can thus be formulated as

$$\min_{f\in\mathcal{H}, \ \alpha\in\mathbb{R}^n} \ \frac{1}{n}\sum_{i=1}^{n} \xi_S(c_i - \alpha_i f(x_i)) + \alpha_i z^*(f(x_i)) - z^*(c_i) \ . \tag{13}$$

We note that the scalar α_i is specific to observation i, and serves as a multiplier of the prediction $f(x_i)$. Now consider forcing each scalar α_i in (13) to be the *same*, i.e., $\alpha_i = \alpha$ for all $i \in \{1,\ldots,n\}$. Although this assumption is an approximation (and upper bounds (13)), Proposition 5 implies that it is a reasonable one if we take the common multiplier α to be large enough. In fact, Proposition 5 shows that the optimal value of each α_i tends to ∞, and is achieved by a large constant when S is polyhedral. Thus we see that α now plays the role of a parameter that uniformly controls the size of the predictions. Instead, we can assume that the size of the predictions are directly controlled by the size of f by setting α as a fixed value, say $\alpha = 2$, which yields the upper bound in (7). Note that setting $\alpha = 2$ may also be interpreted as a change of variables $\tilde{f} := \frac{\alpha}{2} f$. (We note that the choice of $\alpha = 2$ is not yet apparent but is somewhat analogous to including the $\frac{1}{2}$ constant in the least squares loss function. This will be made clear in Section 4.)

PROPOSITION 5. *For any $\hat{c}, c \in \mathbb{R}^d$, the SPO loss function may be expressed as*

$$\ell_{\text{SPO}}(\hat{c}, c) \;=\; \lim_{\alpha \to +\infty} \{\xi_S(c - \alpha\hat{c}) + \alpha z^*(\hat{c})\} - z^*(c) \;. \tag{14}$$

Moreover, if S is polyhedral then the minimum in (10) *is attained, i.e, there exists $\bar{\alpha} \geq 0$ such that for all $\alpha \geq \bar{\alpha}$ the SPO loss function may be expressed as*

$$\ell_{\text{SPO}}(\hat{c}, c) \;=\; \xi_S(c - \alpha\hat{c}) + \alpha z^*(\hat{c}) - z^*(c) \;. \tag{15}$$

Proof. It suffices to show that the function $q(\alpha) = \xi_S(c - \alpha\hat{c}) + \alpha z^*(\hat{c})$ is monotone decreasing on $\mathbb{R}$, from which (14) follows from the basic monotone convergence theorem. Clearly $q(\cdot)$ is a convex function and moreover a subgradient g of $q(\cdot)$ for any α is given by $g := z^*(\hat{c}) - \hat{c}^T w^*(\alpha\hat{c} - c)$. Since $z^*(\hat{c}) = \min_{w \in S} \hat{c}^T w$, we have that $g \leq 0$ for any α. Now, for any $\alpha' \leq \alpha$, the subgradient inequality implies that:

$$q(\alpha') \;\geq\; q(\alpha) + g \cdot (\alpha' - \alpha) \;\geq\; q(\alpha) \;,$$

since g and $\alpha' - \alpha$ are both nonpositive. Thus, $q(\cdot)$ is monotone decreasing.

The representation of (15) follows from a similar proof of Proposition 10, but uses a direct application of a different strong duality result (see, for example, Proposition 5.2.1 of (Bertsekas 1999)) that exploits the assumption that S is polyhedral to additionally obtain both primal and dual attainment. □

Justification of upper bound (8). The final step in the derivation of our convex surrogate SPO+ loss function involves approximating the concave (therefore nonconvex) function $z^*(\cdot)$ with a first-order expansion. Namely, we apply the bound $z^*(\hat{c}) \leq \hat{c}^T w^*(c)$, which can be viewed as a first-order approximation of $z^*(\hat{c})$ based on a supergradient computed at c (i.e., it holds that $w^*(c) \in \partial z^*(c)$). The first-order approximation of $z^*(\hat{c})$ can be intuitively justified since one might expect $w^*(c)$, the optimal decision for the true cost vector, to be a near-optimal solution to $z^*(\hat{c})$, the nominal problem under the predicted cost vector. If the prediction is reasonably close to the true cost vector, then this upper bound is indeed reasonable. In fact, Section 4 provides a consistency property suggesting that the predictions $\hat{c}$ are indeed reasonably close to the true value c if the prediction model is trained on a sufficiently large dataset.

4. Consistency of the SPO+ Loss Function

In this section, we now assume full knowledge of the true underlying distribution of (x, c), and prove a key consistency condition to describe when minimizing the SPO+ loss is equivalent to minimizing the SPO loss. This result is analagous to the well-known consistency results of the hinge loss and logistic loss functions with respect to the 0-1 loss – minimizing hinge and logistic loss under full knowledge also minimizes the 0-1 loss – and provides theoretical motivation for their success in practice.

We let $\mathcal{D}$ denote the distribution of (x, c), i.e., $(x, c) \sim \mathcal{D}$, and consider the population version of the true SPO risk minimization problem:

$$\min_f \; R_{\text{SPO}}(f) := \mathbb{E}_{(x,c)\sim\mathcal{D}}[\ell_{\text{SPO}}(f(x), c)] \;, \tag{16}$$

and the population version of the SPO+ risk minimization problem:

$$\min_f \; R_{\text{SPO+}}(f) := \mathbb{E}_{(x,c)\sim\mathcal{D}}[\ell_{\text{SPO+}}(f(x), c)] \;. \tag{17}$$

Note here that we place no restrictions on $f(\cdot)$, meaning $\mathcal{H}$ consists of any function mapping features to cost vectors. We let f^*_{SPO} denote any optimal solution of (16) and let $f^*_{\text{SPO+}}$ denote any optimal solution of (17). The goal of this section is to demonstrate that the SPO+ loss function is *Fisher consistent* with respect to the SPO loss, i.e., $f^*_{\text{SPO+}}$ is also an optimal solution of the population version of the true SPO risk minimization problem (16).

Throughout this section, we consider a non-parametric setup where the dependence on the features x is dropped without loss of generality. To see this, first observe that $R_{\text{SPO}}(f) =$

$\mathbb{E}_x\left[\mathbb{E}_c\left[\ell_{\text{SPO}}(f(x),c) \mid x\right]\right]$ and likewise for the SPO+ risk. Since there is no constraint on $f(\cdot)$ (the hypothesis class consists of all prediction models), then solving problems (16) and (17) is equivalent to optimizing each $f(x)$ individually for all $x \in \mathcal{X}$. Therefore, for the remainder of the section unless otherwise noted, we drop the dependence on x. Thus, we now assume that the distribution $\mathcal{D}$ is only over c, and the SPO and SPO+ risk is defined as $R_{\text{SPO}}(\hat{c}) := \mathbb{E}_c[\ell_{\text{SPO}}(\hat{c},c)]$ and $R_{\text{SPO+}}(\hat{c}) := \mathbb{E}_c[\ell_{\text{SPO+}}(\hat{c},c)]$, respectively. Moreover, f^*_{SPO} and $f^*_{\text{SPO+}}$ can now each be described as a single vector.

Next, we fully characterize the minimizers of the true SPO risk problem (16) in this setting. For convenience, let us define $\bar{c} := \mathbb{E}_c[c]$ (note that we are implicitly assuming that $\bar{c}$ is finite). Proposition 6 demonstrates that for any minimizer c^* of $R_{\text{SPO}}(\cdot)$, all of its corresponding solutions with respect to the nominal problem, $W^*(c^*)$, are also optimal solutions for $P(\bar{c})$. In other words, minimizing the true SPO risk also optimizes for the expected cost in the nominal problem (since the objective function is linear). Proposition 6 also demonstrates that the converse is true – namely any cost vector prediction with a unique optimal solution that also optimizes for the expected cost is also a minimizer of the true SPO risk.

Proposition 6. *If a cost vector c^* is a minimizer of $R_{\text{SPO}}(\cdot)$, then $W^*(c^*) \subseteq W^*(\bar{c})$. Conversely, if c^* is a cost vector such that $W^*(c^*)$ is a singleton and $W^*(c^*) \subseteq W^*(\bar{c})$, then c^* is a minimizer of $R_{\text{SPO}}(\cdot)$.*

Proof. Consider a cost vector c^* that is a minimizer of $R_{\text{SPO}}(\cdot)$. Let $\bar{w}$ be an optimal solution of $z(\bar{c})$, i.e., $\bar{w} \in W^*(\bar{c})$, and let $\tilde{c}$ be chosen such that $\bar{w}$ is the unique optimal solution of $z(\tilde{c})$, i.e., $W^*(\tilde{c}) = \{\bar{w}\}$. (Note that if $\bar{w}$ is the unique optimal solution of $z(\bar{c})$ then it suffices to select $\tilde{c} \leftarrow \bar{c}$, otherwise we may take $\tilde{c}$ as a slight perturbation of $\bar{c}$). Then it holds that:

$$0 \le R_{\text{SPO}}(\tilde{c}) - R_{\text{SPO}}(c^*) \;=\; \mathbb{E}_c\left[\max_{w \in W^*(\tilde{c})}\left\{c^T w\right\}\right] - \mathbb{E}_c\left[\max_{w \in W^*(c^*)}\left\{c^T w\right\}\right]$$
$$= \mathbb{E}_c[c^T \bar{w}] - \mathbb{E}_c\left[\max_{w \in W^*(c^*)}\left\{c^T w\right\}\right] \;=\; \bar{c}^T \bar{w} - \mathbb{E}_c\left[\max_{w \in W^*(c^*)}\left\{c^T w\right\}\right] \;\le\; \bar{c}^T \bar{w} - \max_{w \in W^*(c^*)}\left\{\bar{c}^T w\right\} \;,$$

where the first inequality and first equality are by definition, the second equality uses $W^*(\tilde{c}) = \{\bar{w}\}$, the third equality uses linearity of expectation, and the final inequality is Jensen's inequality. Finally, we conclude that, for any $w \in W^*(c^*)$, it holds that $\bar{c}^T w \le \bar{c}^T \bar{w} = z^*(\bar{c})$. Therefore, $W^*(c^*) \subseteq W^*(\bar{c})$.

To prove the other direction, consider a cost vector c^* such that $W^*(c^*) = \{w^*(c^*)\}$ is a singleton and $W^*(c^*) \subseteq W^*(\bar{c})$, i.e., $w^*(c^*) \in W^*(\bar{c})$. Let c^{**} be an arbitrary minimizer of $R_{\text{SPO}}(\cdot)$. Then,

$$\begin{aligned} R_{\text{SPO}}(c^*) - R_{\text{SPO}}(c^{**}) &= \mathbb{E}_c\left[\max_{w \in W^*(c^*)} \left\{c^T w\right\}\right] - \mathbb{E}_c\left[\max_{w \in W^*(c^{**})} \left\{c^T w\right\}\right] \\ &= \bar{c}^T w^*(c^*) - \mathbb{E}_c\left[\max_{w \in W^*(c^{**})} \left\{c^T w\right\}\right] \;=\; z^*(\bar{c}) - \mathbb{E}_c\left[\max_{w \in W^*(c^{**})} \left\{c^T w\right\}\right] \\ &\le z^*(\bar{c}) - \max_{w \in W^*(c^{**})} \left\{\bar{c}^T w\right\} \;\le\; 0 \;, \end{aligned}$$

where the second equality uses $W^*(c^*) = \{w^*(c^*)\}$, the third equality uses $w^*(c^*) \in W^*(\bar{c})$, and the first inequality is Jensen's inequality. Finally, we conclude that since c^{**} is a minimizer of $R_{\text{SPO}}(\cdot)$ and c^* has at most the same risk, then c^* is also a minimizer of $R_{\text{SPO}}(\cdot)$. □

Example 2 below demonstrates that, in order to ensure that c^* is a minimizer of $R_{\text{SPO}}(\cdot)$, it is not sufficient to allow c^* to be any cost vector such that $W^*(c^*) \subseteq W^*(\bar{c})$. In fact, it may not be sufficient for c^* to be $\bar{c}$. This follows from the unambiguity of the SPO loss function, which chooses a worst-case optimal solution in the event that the prediction allows for more than one optimal solution.

Example 2. Suppose that $d = 1$, $S = [-1/2, +1/2]$, and c is normally distributed with mean $\bar{c} = 0$ and variance 1. Then $W^*(\bar{c}) = S$ and $\ell_{\text{SPO}}(\bar{c}, c) = \xi_S(c) - z^*(c) = |c|$ for all c. Clearly though, $\ell_{\text{SPO}}(1, c) = -c/2 + |c|/2$. Moreover, $\ell_{\text{SPO}}(1, c)$ strictly dominates $\ell_{\text{SPO}}(\bar{c}, c)$ in the sense that $\ell_{\text{SPO}}(1, c) \le \ell_{\text{SPO}}(\bar{c}, c)$ for all c and $\ell_{\text{SPO}}(1, c) < \ell_{\text{SPO}}(\bar{c}, c)$ for $c > 0$. Therefore, $R_{\text{SPO}}(1) < R_{\text{SPO}}(\bar{c})$ and hence $\bar{c}$ is not a minimizer of $R_{\text{SPO}}(\cdot)$. □

Note that Proposition 6 implies that if $W^*(\bar{c})$ is unique, then $\bar{c}$ is a minimizer of the SPO loss. In fact, $\bar{c}$ is the minimizer of the least squares loss under the population setting, making least squares consistent with the SPO loss. We next show that the SPO+ loss function is also consistent under mild conditions, and show that SPO+ is also empirically superior in Section 6.

4.1. Consistency of the SPO+ Estimator

We now examine Fisher consistency of the SPO+ loss function, which implies that minimizing the SPO+ risk (17) also minimize the SPO risk (16). Recall that the expected risk for the SPO+ loss is defined as $R_{\text{SPO+}}(\hat{c}) := \mathbb{E}_c[\ell_{\text{SPO+}}(\hat{c}, c)]$. It turns out that the SPO+ loss function is not always consistent with respect to the SPO loss, i.e., it is possible that a minimizer of $R_{\text{SPO+}}(\cdot)$ may be strictly suboptimal for the problem of minimizing the true risk $R_{\text{SPO}}(\cdot)$.

Assumption 1 presents a set of natural sufficient conditions on the distribution of c that lead to Fisher consistency of the SPO+ estimator.

ASSUMPTION 1. *The following are conditions that imply Fisher consistency of the SPO+ loss function.*

1. *The distribution of c is continuous on all of $\mathbb{R}^d$.*
2. *The distribution of c is centrally symmetric about its mean $\bar{c}$.*
3. *The mean $\bar{c}$ has a unique optimal solution, i.e., $W^*(\bar{c})$ is a singleton.*
4. *The feasible region S is not a singleton.*

More formally, "continuous on all of $\mathbb{R}^d$" means that c has a probability density function that is strictly greater than 0 at every point in $\mathbb{R}^d$ and "centrally symmetric about its mean" means that $c - \bar{c}$ is equal in distribution to $\bar{c} - c$. The Gaussian distribution with any mean and covariance matrix is an example which satisfies Assumption 1. (Note that this is a typical assumption when the true model is assumed to have the form $c = f(x) + \epsilon$, where ϵ is a Gaussian random variable.) Requiring $\bar{c}$ to have a unique optimizer with respect to the nominal problem is a minimal assumption, as the measure of the space of vectors with multiple optimizers is minimal. Finally, the case where S is a singleton is actually trivial since every function is a minimizer of the true SPO risk.

Under these conditions, Theorem 1 shows that $\bar{c}$ is the unique minimizer of $R_{\text{SPO+}}(\cdot)$, which, by Proposition 6, implies that minimizing $R_{\text{SPO+}}(\cdot)$ also minimizes $R_{\text{SPO}}(\cdot)$. Thus, under Assumption 1, the SPO+ loss function is Fisher consistent with respect to the SPO loss function.

THEOREM 1 **(Consistency of SPO+)**. *Suppose Assumption 1 holds. Then there is a unique global minimizer c^* of $R_{\text{SPO+}}(\cdot)$. Moreover, $c^* = \bar{c}$ and thus c^* also minimizes $R_{\text{SPO}}(\cdot)$.*

Proof. The proof works in two steps. First, we show that $\bar{c}$ is an optimal solution of $\min_{\hat{c} \in \mathbb{R}^d} R_{\text{SPO+}}(\hat{c})$ by considering the optimality conditions of this problem. Second, we directly show that $\bar{c}$ is the unique such minimizer. Since $W^*(\bar{c})$ is assumed to be a singleton, then Proposition 6 implies that $\bar{c}$ also minimizes $R_{\text{SPO}}(\cdot)$.

Step 1: First, note that it is straightforward to show that $R_{\text{SPO+}}(\hat{c})$ is finite valued for all $\hat{c}$, which follows since S is compact and $\bar{c} := \mathbb{E}[c]$ is finite. Moreover, by Proposition 2, it is clear that $R_{\text{SPO+}}(\cdot)$ is convex on $\mathbb{R}^d$. In particular, for any point $\hat{c}$ the subdifferential $\partial R_{\text{SPO+}}(\hat{c})$ is nonempty and, since $R_{\text{SPO+}}(\hat{c})$ is finite, we have that $\partial R_{\text{SPO+}}(\hat{c}) = \mathbb{E}_c [\partial \ell_{\text{SPO+}}(\hat{c}, c)]$ (see

Strassen (1965), Bertsekas (1973)), where the latter refers to the selection expectation over the random set $\partial\ell_{\text{SPO+}}(\hat{c}, c)$ (see Molchanov (2005)). By the linearity of selection expectation, note that $\partial R_{\text{SPO+}}(\hat{c}) = -2\mathbb{E}_c[W^*(2\hat{c} - c)] + 2\mathbb{E}_c[w^*(c)] = -2\mathbb{E}_c[w^*(2\hat{c} - c)] + 2\mathbb{E}_c[w^*(c)]$, where the second equality follows since the distribution of c is continuous on all of $\mathbb{R}^d$ which implies that $W^*(2\hat{c} - c)$ is a singleton with probability 1 (see, e.g., the introductory discussion in Drusvyatskiy and Lewis (2011)).

Now, the optimality conditions for the convex problem $\min_{\hat{c}\in\mathbb{R}^d} R_{\text{SPO+}}(\hat{c})$ state that c^* is a global minimizer if and only if $0 \in \partial R_{\text{SPO+}}(c^*)$. By the discussion in the previous paragraph, the optimality conditions may be equivalently written as $\mathbb{E}_c[w^*(c)] = \mathbb{E}_c[w^*(2c^* - c)]$. Finally, since c is centrally symmetric around its mean, we have that c is equal in distribution to $2\bar{c} - c$; hence $\mathbb{E}_c[w^*(c)] = \mathbb{E}_c[w^*(2\bar{c} - c)]$. Therefore $\bar{c}$ satisfies $0 \in \partial R_{\text{SPO+}}(\bar{c})$ and is an optimal solution of $\min_{\hat{c}\in\mathbb{R}^d} R_{\text{SPO+}}(\hat{c})$.

Step 2: Consider any vector $\Delta \neq 0$, and let us rewrite the difference $R_{\text{SPO+}}(\bar{c} + \Delta) - R_{\text{SPO+}}(\bar{c})$ as follows:

$$
\begin{aligned}
R_{\text{SPO+}}(\bar{c} + \Delta) - R_{\text{SPO+}}(\bar{c}) &= \mathbb{E}_c[\xi_S(c - 2(\bar{c} + \Delta)) + 2(\bar{c} + \Delta)^T w^*(c) - z^*(c)] - \\
&\quad\ \mathbb{E}_c[\xi_S(c - 2\bar{c}) + 2\bar{c}^T w^*(c) - z^*(c)] \\
&= \mathbb{E}_c[\xi_S(c - 2\bar{c} - 2\Delta) - \xi_S(c - 2\bar{c}) + 2\Delta^T w^*(c)] \\
&= \mathbb{E}_c[\xi_S(c - 2\bar{c} - 2\Delta) - (c - 2\bar{c})^T w^*(2\bar{c} - c) + 2\Delta^T w^*(c)] \\
&= \mathbb{E}_c[\xi_S(c - 2\bar{c} - 2\Delta) - (c - 2\bar{c} - 2\Delta)^T w^*(2\bar{c} - c)] \\
&= \mathbb{E}_c[(c - 2\bar{c} - 2\Delta)^T (w^*(2\bar{c} + 2\Delta - c) - w^*(2\bar{c} - c))] .
\end{aligned}
$$

The first equality above uses the definitions of $R_{\text{SPO+}}(\cdot)$ and $\ell_{\text{SPO+}}(\cdot,\cdot)$, the second uses linearity of expectation, the third uses the definition of $\xi_S(\cdot)$, the fourth uses the optimality conditions $\mathbb{E}_c[w^*(c)] = \mathbb{E}_c[w^*(2\bar{c} - c)]$ and linearity of expectation, and the fifth uses the definition of $\xi_S(\cdot)$. Since $w^*(2\bar{c} + 2\Delta - c)$ is the maximizer for $c - 2\bar{c} - 2\Delta$, we have that $(c - 2\bar{c} - 2\Delta)^T(w^*(2\bar{c} + 2\Delta - c) - w^*(2\bar{c} - c))$ is a nonnegative random variable. Moreover, since the distribution of c is continuous on all of $\mathbb{R}^d$ and S is not a singleton, it holds that $\mathbb{P}(w^*(2\bar{c} + 2\Delta - c) \neq w^*(2\bar{c} - c)) > 0$ and note also that $w^*(2\bar{c} + 2\Delta - c)$ is the unique maximizer for $c - 2\bar{c} - 2\Delta$ with probability one. Thus, we have $\mathbb{P}((c - 2\bar{c} - 2\Delta)^T (w^*(2\bar{c} + 2\Delta - c) - w^*(2\bar{c} - c)) > 0) > 0$ and therefore

$$
R_{\text{SPO+}}(\bar{c} + \Delta) - R_{\text{SPO+}}(\bar{c}) = \mathbb{E}_c[(c - 2\bar{c} - 2\Delta)^T (w^*(2\bar{c} + 2\Delta - c) - w^*(2\bar{c} - c))] > 0 .
$$

Thus $\bar{c} + \Delta$ is not a minimizer of the SPO risk, and $\bar{c}$ must be the unique minimizer. $\square$

As mentioned previously, the minimizer for the least squares (LS) loss function is also $\bar{c}$, and thus the least squares loss is also Fisher consistent with respect to the SPO loss. A priori, one cannot claim LS or SPO+ to be better than the other, but we now derived a brand new loss function, SPO+, that maintains a fundamental consistency property of the de facto standard LS loss function. Since this consistency property only holds under full information and no model misspecification, numerical testing is needed to demonstrate that SPO+ indeed outperforms least squares, primarily due to its well-motivated derivation.

We also remark that the first two, and most important, conditions of Assumption 1 are not individually sufficient to ensure consistency on their own. Example 3 demonstrates a situation where the distribution of c is symmetric about its mean but there exists a minimizer of the SPO+ risk that does not minimize the SPO risk. Example 4 demonstrates a situation where c is continuous on $\mathbb{R}^d$ and the minimizer of SPO+ is unique, but it does not minimize the SPO risk. Note also that Example 2 in the previous subsection demonstrates a situation where all of the conditions of Assumption 1 are satisfied except the mean $\bar{c}$ does not have a unique optimal solution. Hence, for this example, $R_{\text{SPO+}}(\cdot)$ has a unique optimal solution given by $\bar{c}$ but $\bar{c}$ does not minimize $R_{\text{SPO}}(\cdot)$.

Example 3. Define S as the two-dimensional simplex with extreme points at $(0,0),(1,0)$, and $(0,1)$. Let c have a two point distribution on the points $(-2,1)$ and $(0,-1)$, each with probability of 0.5. One can confirm that $\bar{c}=(-1,0)$, $W^*(\bar{c})=\{(1,0)\}$, $\mathbb{E}_c[w^*(c)]=(0.5,0.5)$, and that c is symmetric around its mean (c is equal in distribution to $2\bar{c}-c$).

Now we claim $c^*=(-0.25,-0.5)$ is a minimizer of SPO+ risk. This is confirmed by checking that $\mathbb{E}_c[w^*(2c^*-c)]=(0.5,0.5)$. However, $W^*(c^*)=\{(0,1)\}\not\subseteq\{(1,0)\}=W^*(\bar{c})$. Thus by Proposition 6, c^* is not a minimizer of the SPO risk. □

Example 4. Define S as the two-dimensional unit square with extreme points at $(0,0),(1,0),(1,1)$ and $(0,1)$. Let c have a continuous distribution on $\mathbb{R}^d$ defined in the following manner. The probability that c occurs in each quadrant is exactly 0.25. The distribution over each quadrant is a folded normal. The mean of the folded normals is $(9,9)$ in quadrant 1, $(-1,1)$ in quadrant 2, $(-1,-1)$ in quadrant 3, and $(1,-1)$ in quadrant 4. Thus, $\bar{c}=(2,2)$ and $W^*(\bar{c})=\{(0,0)\}$. Note that all cost vectors in quadrant 1 are minimized by $(0,0)$, quadrant 2 by $(1,0)$, quadrant 3 by $(1,1)$, and quadrant 4 by $(0,1)$. Therefore, $\mathbb{E}_c[w^*(c)]=(0.5,0.5)$.

Now we claim $c^*=(0,0)$ is the unique minimizer of SPO+ risk. Since the $w^*(c)$ only depends on which quadrant c lies in and both c and $-c$ lie in each of the four quadrants with equal probability, it holds that $\mathbb{E}_c[w^*(2c^* - c)] = \mathbb{E}_c[w^*(-c)] = \mathbb{E}_c[w^*(c)] = (0.5, 0.5)$ which implies that c^* is optimal. Finally, observe that $W^*(c^*) = S \not\subseteq \{(0,0)\} = W^*(\bar{c})$, which means that by Proposition 6, c^* is not a minimizer of the SPO risk. □

5. Computational Approaches

In this section, we consider computational approaches for solving the SPO problem (9). Herein, we focus on the case of linear predictors, $\mathcal{H} = \{f : f(x) = Bx \text{ for some } B \in \mathbb{R}^{d\times p}\}$, with regularization possibly incorporated into the objective function, using the regularizer $\Omega(\cdot) : \mathbb{R}^{d\times p} \to \mathbb{R}$. (This is equivalent to working with the hypothesis class $\mathcal{H} = \{f : f(x) = Bx \text{ for some } B \in \mathbb{R}^{d\times p}, \Omega(B) \leq \rho\}$ for some $\rho > 0$.) For example, we may use the ridge penalty $\Omega(B) = \frac{1}{2}\|B\|_F^2$, where $\|B\|_F$ denotes the Frobenius norm of B, i.e., the entry-wise ℓ_2 norm. Other possibilities include an entry-wise ℓ_1 penalty or the nuclear norm penalty, i.e., an ℓ_1 penalty on the singular values of B. In any case, these presumptions lead to the following version of (9):

$$\min_{B\in\mathbb{R}^{d\times p}} \ \frac{1}{n}\sum_{i=1}^{n} \ell_{\text{SPO+}}(Bx_i, c_i) \ + \lambda\Omega(B) \ , \tag{18}$$

where $\lambda \geq 0$ is a regularization parameter. Since the SPO loss is convex as stated in Proposition 2, then the above problem is a convex optimization problem as long as $\Omega(\cdot)$ is a convex function. We mainly consider two approaches for solving problem (18): *(i)* reformulations based on modeling $\ell_{\text{SPO+}}(\cdot, c)$ using duality, and *(ii)* stochastic gradient based methods that instead rely only on an optimization oracle for problem (2). The reformulation based approach *(i)* requires an explicit description of the feasible region S, for example if S is a polytope then this approach necessitates working with an explicit list of inequality constraints describing S. On the other hand, the stochastic gradient based approach *(ii)* does not require an explicit description of S and instead works by iteratively calling the optimization oracle $w^*(\cdot)$. Therefore it is much more straightforward to adapt the stochastic gradient descent approach to problems with complicated constraints, such as mixed integer or nonlinear problems. While approach *(i)* is more restrictive in its requirements, it does offer a few advantages. Depending on the structure of S, for example if S is a polytope with known linear inequality constraints, then approach *(i)* may able to utilize off-the-shelf conic optimization solvers such as CPLEX and Gurobi that are capable of producing high accuracy solutions for small to medium sized problem instances. However, for large scale instances where d, p, and n might be very large, conic solvers based on interior point methods do not scale as well. Stochastic gradient methods, on the other hand, scale much better to instances where n may be extremely large, and possibly also to instances where d and p are large but the optimization oracle $w^*(\cdot)$ is efficiently computable due to the speical structure of S.

5.1. Reformulation Approach

Let us first discuss the reformulation approach *(i)*, which aims to recast problem (18) in a form that is amenable to popular optimization solvers. To describe this approach, we presume that S is a polytope described by known linear inequalities, i.e., $S = \{w : Aw \geq b\}$ for some given problem data $A \in \mathbb{R}^{m\times d}$ and $b \in \mathbb{R}^m$. The same approach may also be applied to particular classes of nonlinear feasible regions, although the complexity of the resulting reformulated problem will be different. The key idea is that when S is a polytope, then $\ell_{\text{SPO+}}(\cdot, c)$ is a (piecewise linear) convex function of the prediction $\hat{c}$ and therefore the epigraph of $\ell_{\text{SPO+}}(\cdot, c)$ can be tractably modeled with linear constraints by employing linear programming duality. Proposition 7 formalizes this approach. (Recall that, for $w \in \mathbb{R}^d$ and $x \in \mathbb{R}^p$, wx^T denotes $d \times p$ outer product matrix where $(wx^T)_{ij} = w_i x_j$.)

PROPOSITION 7. *The regularized SPO problem* (18) *is equivalent to the following optimization problem:*

$$\begin{array}{lll} \min\limits_{B,p} & \dfrac{1}{n}\displaystyle\sum_{i=1}^{n}\left[-b^T p_i + 2(w^*(c_i)x_i^T)\bullet B - z^*(c_i)\right] \;+\; \lambda\Omega(B) & \\ \text{s.t.} & A^T p_i = 2Bx_i - c_i & \text{for all } i \in \{1,\ldots,n\} \\ & p_i \in \mathbb{R}^m, p_i \geq 0 & \text{for all } i \in \{1,\ldots,n\} \\ & B \in \mathbb{R}^{d\times p} \;. & \end{array} \tag{19}$$

Proof. Recall that $\ell_{\text{SPO+}}(Bx_i, c_i) = \xi_S(c_i - 2Bx_i) + 2(Bx_i)^T w^*(c_i) - z^*(c_i)$. Linear programming strong duality implies that:

$$\begin{array}{rclcl} \xi_S(c_i - 2Bx_i) & = & \max\limits_{w}\ (c_i - 2Bx_i)^T w & = & \min\limits_{p}\ -b^T p_i \\ & & \text{s.t.}\quad Aw \geq b & & \text{s.t.}\ \ -A^T p = c_i - 2Bx_i \\ & & & & \qquad p \geq 0\ . \end{array}$$

Note also that $2(w^*(c_i)x_i^T) \bullet B$ is just a rewriting of $2(Bx_i)^T w^*(c_i)$ as an explicit linear function. Thus, introducing variables $p_i \in \mathbb{R}^m$ for each $i \in \{1,\ldots,n\}$, it is clear that (19) is equivalent to (18). □

Thus, as we can see, problem (19) is almost a linear optimization problem – the only part that may be nonlinear is the regularizer $\Omega(\cdot)$. For several natural choices of $\Omega(\cdot)$, problem (7) may be cast as a conic optimization problem that can be solved efficiently with interior point methods. Proposition 8 summarizes results for three common regularizers.

PROPOSITION 8. *Consider the equivalent form of* (18) *given in* (19)*. Then, it holds that:*

1. *If* $\Omega(\cdot)$ *is the entrywise* ℓ_1*-norm,* $\Omega(B) = \|B\|_1$*, then* (19) *is equivalent to a linear programming (LP) problem.*
2. *If* $\Omega(\cdot)$ *is the ridge penalty,* $\Omega(B) = \frac{1}{2}\|B\|_F^2$*, then* (19) *is equivalent to quadratic programming (QP) problem.*
3. *If* $\Omega(\cdot)$ *is the nuclear norm penalty,* $\Omega(B) = \|B\|_*$*, then* (19) *is equivalent to a semidefinite programming (SDP) problem.*

5.2. Stochastic Gradient Approach

The idea of this approach is to apply a stochastic gradient method (see, for example, (Robbins and Munro 1951), Bottou et al. (2016), and the references therein) directly to problem (18) in its original format. For this approach, we assume that the regularizer is convex and differentiable. For example, the ridge penalty $\Omega(B) = \frac{1}{2}\|B\|_F^2$ satisfies this assumption. Extensions to other non-differentiable regularizers, such as the entrywise ℓ_1 or nuclear norm penalties, may be done by using a subgradient oracle for $\Omega(\cdot)$ but are typically best handled with the use of more intricate proximal gradient type methods.

Let us now describe some more details about how to apply this approach. For convenience, let us write the objective function of (18) as $L^n_{\text{SPO+}}(B) := \frac{1}{n}\sum_{i=1}^n \phi_i(B)$, where $\phi_i(B) := \ell_{\text{SPO+}}(Bx_i, c_i) + \lambda\Omega(B)$. As mentioned in Proposition 2, $\ell_{\text{SPO+}}(\cdot, c)$ is convex (but not necessarily differentiable) for a fixed c and therefore $\phi_i(\cdot)$ is equal to the sum of a non-differentiable convex function and a differentiable convex function. To compute subgradients of $\phi_i(\cdot)$, recall from the discussion in Section 3 that $2(w^*(c) - w^*(2\hat{c} - c)) \in \partial\ell_{\text{SPO+}}(\hat{c}, c)$ and therefore for any B, B' it holds that:

$$
\begin{aligned}
\ell_{\text{SPO+}}(B'x_i, c_i) \;\geq\; & \ell_{\text{SPO+}}(Bx_i, c_i) \;+\; 2(w^*(c_i) - w^*(2Bx_i - c_i))^T(B'x_i - Bx_i) \\
=\; & \ell_{\text{SPO+}}(Bx_i, c_i) + (2(w^*(c_i) - w^*(2Bx_i - c_i))x_i^T) \bullet (B' - B) \;,
\end{aligned}
$$

and hence $2(w^*(c_i) - w^*(2Bx_i - c_i))x_i^T \in \partial\ell_{\text{SPO+}}(Bx_i, c_i)$ and $2(w^*(c_i) - w^*(2Bx_i - c_i))x_i^T + \lambda\nabla\Omega(B) \in \partial\phi_i(B)$.

Algorithm 1 presents the application of stochastic gradient descent with mini-batching to problem (18). Algorithm 1 is a standard application of stochastic subgradient descent in our setting, and closely follows Nemirovski et al. (2009). It is important to emphasize that the main computational requirement of Algorithm 1 is access to the optimization oracle $w^*(\cdot)$ for problem (2), which is utilized N times during each iteration of the method. The main parameters that need to be set are the batch size parameter N and the step-size sequence $\{\gamma_t\}$ (in addition to the regularization parameter λ). In Section 6, we describe precisely how we set these parameters for our experiments. In general, we recommend setting the batch size parameter to a fixed constant such as 5 or 10. The choice of the step-size depends on the properties of the regularizer $\Omega(\cdot)$. For general convex $\Omega(\cdot)$, or simply when $\lambda = 0$, we recommend following Nemirovski et al. (2009) where it is suggested to set the step-size sequence to $\gamma_t = \frac{\theta}{\sqrt{t+1}}$ for a fixed constant $\theta > 0$. In the case of the ridge penalty $\Omega(B) = \frac{1}{2}\|B\|_F^2$ and when $\lambda > 0$, since this function is strongly convex one may alternatively set the step-size sequence to $\gamma_t = \frac{2}{\lambda(t+2)}$. Since the SPO+ loss function is convex but non-smooth (and also Lipschitz continuous), these are essentially the only two options that will lead to a precise convergence rate guarantee. Indeed, for the sequence $\gamma_t = \frac{\theta}{\sqrt{t+1}}$, Nemirovski et al. (2009) derive an $O(1/\epsilon^2)$ complexity guarantee in terms of the number of iterations required to ensure that the averaged iterate is ϵ-suboptimal in expectation. It is also straightforward to see that this guarantee can be extended to cases when the optimization oracle $w^*(\cdot)$ is only computed approximately with an additive approximation error, as long as we only desire convergence on the order of the accuracy of the oracle. On the other hand, Lacoste-Julien et al. (2012) derive a similar $O(1/(\lambda\epsilon))$ complexity guarantee for the $\gamma_t = \frac{2}{\lambda(t+2)}$ sequence.

It is important to note that Algorithm 1 is a standard and basic application of stochastic subgradient descent, and that one may consider making several adjustments to the method in order to improve its practical and possibly theoretical performance (see, e.g., Bottou (2012), and Bottou et al. (2016) and the references therein). We also mention that one may employ early stopping with this method, whereby we maintain a validation set to monitor (estimates of) the out-of-sample *true SPO loss* as the algorithm runs and then ultimately choose the averaged iterate $\bar{B}_t$ with the smallest true SPO loss ℓ_{SPO} on the validation set as the final model.

Algorithm 1 Stochastic Gradient Descent with mini-batching for problem (18)

Initialize $B_0 \in \mathbb{R}^{d\times p}$ (typically $B_0 \leftarrow 0$), $t \leftarrow 0$. Set batch size parameter $N \geq 1$.

At iteration $t \geq 0$:

1. For $j = 1, \ldots, N$:

   Sample i uniformly at random from the set $\{1, \ldots, n\}$.

   Compute $\tilde{w}_t^j \leftarrow w^*(2B_t x_i - c_i)$.

   Set $\tilde{G}_t^j \leftarrow (w^*(c_i) - \tilde{w}_t^j)x_i^T + \lambda \nabla\Omega(B_t)$.

2. Select $\gamma_t > 0$ and set:

   $G_t \leftarrow \frac{1}{N}\sum_{j=1}^{n} \tilde{G}_t^j$

   $B_{t+1} \leftarrow B_t - \gamma_t G_t$

   $\bar{B}_t \leftarrow \frac{1}{\sum_{s=0}^{t}\gamma_s}\sum_{s=0}^{t}\gamma_s B_s$.

6. Computational Experiments

In this section, we present computational results of synthetic data experiments wherein we empirically examine the quality of the SPO+ loss function for training prediction models, using the shortest path problem and portfolio optimization as our exemplary problem classes. Following Section 5, we focus on linear prediction models, possibly with either ridge or entrywise ℓ_1 regularization. We compare the performance of four different methods:

1. the previously described SPO+ method, (18).
2. the least squares method that replaces the SPO+ loss function in (18) with $\ell(\hat{c}, c) = \frac{1}{2}\|\hat{c} - c\|_2^2$ and also uses regularization whenever SPO+ does.
3. an absolute loss function (i.e., ℓ_1) approach that replaces the SPO+ loss function in (18) with $\ell(\hat{c}, c) = \|\hat{c} - c\|_1$ and also uses regularization whenever SPO+ does.
4. a random forests approach that independently trains d different random forest models for each component of the cost vector, using standard parameter settings of $\lceil p/3 \rceil$ random features at each split and 100 trees.

Note that methods (2.), (3.) and (4.) above do not utilize the structure of S in any way and hence may be viewed as independent learning algorithms with respect to each of the components of the cost vector. For methods (1.), (2.), and (3.) above, we include an intercept column in B that is not regularized. In order to ultimately measure and compare the performance of the four different methods, we compute a "normalized" version of the SPO loss of each of the four previously trained models on an independent test set of size $10,000$. Specifically, if $(\tilde{x}_1, \tilde{c}_1), (\tilde{x}_2, \tilde{c}_2), \ldots, (\tilde{x}_{n_{\text{test}}}, \tilde{c}_{n_{\text{test}}})$ denotes the test set, then we define the normalized test SPO loss of a previously trained model $\hat{f}$ by $\text{NormSPOTest}(\hat{f}) := \frac{\sum_{i=1}^{n_{\text{test}}} \ell_{\text{SPO}}(\hat{f}(\tilde{x}_i), \tilde{c}_i)}{\sum_{i=1}^{n_{\text{test}}} z^*(\tilde{c}_i)}$. Note that we naturally normalize by the total optimal cost of the test set given full information, which with high probability will be a positive number for the examples studied herein.

6.1. Shortest Path Problem

We consider a shortest path problem on a 5×5 grid network, where the goal is to go from the northwest corner to the southeast corner and the edges only go south or east. In this case, the feasible region S can be modeled using network flow constraints as in Example 1. We utilize the reformulation approach given by Proposition 7 to solve the SPO+ training problem (18). Specifically, we use the JuMP package in Julia (Dunning et al. (2017)) with the Gurobi solver to implement problem (19). The optimization problems required in methods (2.) and (3.) are also solved directly using Gurobi. In some cases we use ℓ_1 regularization for methods (1.), (2.), and (3.), in which case, in order to tune the regularization parameter λ, we try 10 different values of λ evenly spaced on the logarithmic scale between 10^{-6} and 100. Furthermore, we use a validation set approach where we train the 10 different models on a training set of size n and then use an independent validation set of size $n/4$ to pick the model that performs best with respect to the SPO loss.

Synthetic Data Generation Process. Let us now describe the process used for generating the synthetic experimental data instances for both problem classes. Note that the dimension of the cost vector $d = 40$ corresponds to the total number of edges in the 5×5 grid network and that p is a given number of features. First, we generate a random matrix $B^* \in \mathbb{R}^{d \times p}$ that encodes the parameters of the true model, whereby each entry of B^* is Bernoulli random variable that is equal to 1 with probability 0.5. We generate the training data $(x_1, c_1), (x_2, c_2), \ldots, (x_n, c_n)$ and the testing data $(\tilde{x}_1, \tilde{c}_1), (\tilde{x}_2, \tilde{c}_2), \ldots, (\tilde{x}_n, \tilde{c}_n)$ according to the following generative model:

1. First, the feature vector $x_i \in \mathbb{R}^p$ is generated from a multivariate Gaussian distribution with i.i.d. standard normal entries, i.e., $x_i \sim N(0, I_p)$.
2. Then, the cost vector c_i is generated according to $c_{ij} = \left[\left(\frac{1}{\sqrt{p}}(B^* x_i)_j + 3\right)^{\text{deg}} + 1\right] \cdot \varepsilon_i^j$ for $j = 1, \ldots, d$, and where c_{ij} denotes the j^{th} component of c_i and $(B^* x_i)_j$ denotes the j^{th} component of $B^* x_i$. Here, deg is a fixed positive integer parameter and ε_i^j is a

multiplicative noise term that is generated independently at random from the uniform distribution on $[1-\bar{\varepsilon}, 1+\bar{\varepsilon}]$ for some parameter $\bar{\varepsilon} \geq 0$.

Note that the model for generating the cost vectors employs a polynomial kernel function (see, e.g., Hofmann et al. (2008)), whereby the regression function for the cost vector given the features, i.e., $\mathbb{E}[c \mid x]$, is a polynomial function of x and the parameter deg dictates the degree of the polynomial. Importantly, we still employ a linear hypothesis class for methods (1.)-(3.) above, hence the parameter deg controls the amount of *model misspecification* and as deg increases we expect the performance of the SPO+ approach to improve relative to methods (2.) and (3.). When $\deg = 1$, the expected value of c is indeed linear in x. Furthermore, for large values of deg, the least squares method will be sensitive to outliers in the cost vector generation process, which is our main motivation for also comparing against the absolute loss approach that is less sensitive to outliers. On the other hand, the random forests method is a non-parametric learning algorithm and will accurately learn the regression function for any value of deg. However, the practical performance of random forests depends heavily on the sample size n and, for relatively small values of n, random forests may perform poorly.

Results. In the following set of experiments on the shortest path problem we described, we fix the number of features at $p = 5$ throughout and, as previously mentioned, use a 5×5 grid network, which implies that $d = 40$. Hence, in total there are $pd = 200$ parameters to estimate. We vary the training set size $n \in \{100, 1000, 5000\}$, we vary the parameter $\deg \in \{1, 2, 4, 6, 8\}$, and we vary the noise half-width parameter $\bar{\varepsilon} \in \{0, 0.5\}$. For every value of n, deg, and $\bar{\varepsilon}$, we run 50 simulations, each of which has a different B^* and therefore different ground truth model. For the cases where $n \in \{100, 1000\}$, we employ ℓ_1 regularization for methods (1.)-(3.), as previously described. When $n = 5000$ we do not use any regularization (since it did not appear to provide any value). As mentioned previously, for each simulation, we evaluate the performance of the trained models by computing the normalized SPO loss on a test set of 10,000 samples. Figure 2 summarizes our findings, and note that the box plot for each configuration of the parameters is across the 50 independent trials.

From Figure 2, we can see that for small values of the deg parameter, i.e., $\deg \in \{1, 2\}$, the absolute loss, least squares, and SPO+ methods perform comparably, with the least squares method slightly dominating in the case of noise with $\bar{\varepsilon} = 0.5$. The slight dominance of least squares (and sometimes the absolute loss as well) in these cases might be explained by some inherent robustness properties of the least squares loss. It is also plausible that,

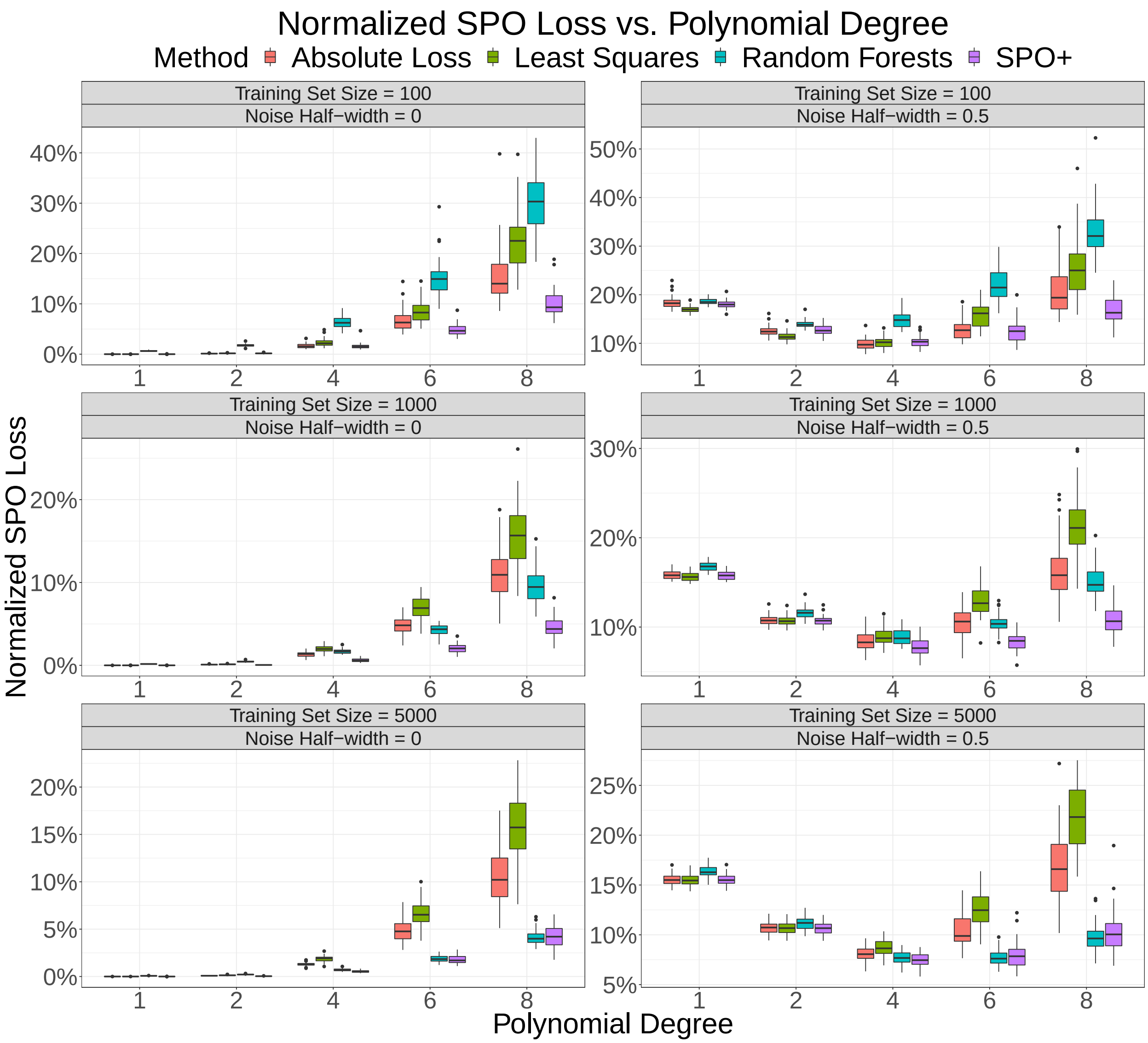


Figure 2 **Normalized test set SPO loss for the SPO+, least squares, absolute loss, and random forests methods on shortest path problem instances.**

since the SPO+ loss function is more intricate than the "simple" least squares loss function, it may overfit in situations with noise and a small training set size. On the other hand, as the parameter deg grows and the degree of model misspecification increases, then the SPO+ approach generally begins to perform best across all instances except when $n = 5000$, in which case random forests performs comparably to SPO+. This behavior suggests that the SPO+ loss is better than the competitors at leveraging additional data and stronger nonlinear signals.

It is interesting to point out that random forests generally does not perform well except when $n = 5000$, in which case it performs comparably to SPO+, which uses a much simpler linear hypothesis class. Indeed, when $n \in \{100, 1000\}$, random forests almost always performs worst, except for when $n = 1000$ and $\deg \in \{6, 8\}$, in which case random forests outperforms least squares, performs comparably to the absolute loss method, and is strongly dominated by SPO+. Indeed, the cases where $n \in \{1000, 5000\}$ and $\deg \in \{6, 8\}$ suggest that least squares is prone to outliers whereas the absolute loss is not, random forests is slow to converge due to its non-parametric nature, and SPO+ is best able to adapt to the large degree of model misspecification even with a modest amount of data (i.e., $n = 1000$).

6.2. Portfolio Optimization

Here we consider a simple portfolio selection problem based on the classical Markowitz model (Markowitz 1952). As discussed in Section 1, we presume that there are auxiliary features that may be used to predict the returns of d different assets, but that the covariance matrix of the asset returns *does not depend* on the auxiliary features. Therefore, we consider a model with a constraint that bounds the overall variance of the portfolio. Specifically, if $\Sigma \in \mathbb{R}^{d \times d}$ denotes the (positive semidefinite) covariance matrix of the asset returns and $\gamma \geq 0$ is the desired bound on the overall variance (risk level) of the portfolio, then the feasible region S in (2) is given by $S := \{w : w^T \Sigma w \leq \gamma, e^T w \leq 1, w \geq 0\}$. Here e denotes the vector of all ones and since we only require that $e^T w \leq 1$, the cost vector c in (2) represents the negative of the incremental returns of the assets above the risk-free rate. In other words, it holds that $c = -\tilde{r}$ where $\tilde{r} = r - r_{\text{RF}} e$, r represents the vector of asset returns, and r_{RF} is the risk-free rate.

In this experiment, we set the number of assets $d = 50$ and the return vectors and features are synthetically generated according to a similar process as before. As before, we generate a random matrix $B^* \in \mathbb{R}^{d \times p}$ that encodes the parameters of the true model, whereby each entry of B^* is Bernoulli random variable that is equal to 1 with probability 0.5. Then, for some noise level parameter $\tau \geq 0$, we generate a factor loading matrix $L \in \mathbb{R}^{50 \times 4}$ such that each entry of L is uniformly distributed on $[-0.0025\tau, 0.0025\tau]$ independently of everything else. To generate a training/testing pair (x_i, c_i), as before we first generate the feature vector $x_i \in \mathbb{R}^p$ from a multivariate Gaussian distribution with i.i.d. standard normal entries, i.e., $x_i \sim N(0, I_p)$. Then, the incremental return vector $\tilde{r}_i$ is generated according to the following process:

1. The conditional mean $\bar{r}_{ij}$ of the j^{th} asset return is set equal to $\bar{r}_{ij} := \left(\frac{0.05}{\sqrt{p}}(B^* x_i)_j + (0.1)^{1/\deg}\right)^{\deg}$, where deg is a fixed positive integer parameter.

2. The observed return vector $\tilde{r}_i$ is set to $\tilde{r}_i := \bar{r}_i + Lf + 0.01\tau\varepsilon$, where $f \sim N(0, I_4)$ and $\varepsilon \sim N(0, I_{50})$. The cost vector c_i is set to $c_i := -\tilde{r}_i$.

It follows from step (2.) above that, conditional on the observed features $x_i \in \mathbb{R}^p$, the covariance matrix of the returns is $\Sigma = LL^T + (0.01\tau)^2 I_{50}$. We set the risk level parameter γ, which is part of the definition of the feasible region S, to $\gamma := 2.25 \cdot \bar{w}^T \Sigma \bar{w}$ where $\bar{w} := e/10$ is the equal weight portfolio. Note that the particular functional form of the conditional mean returns in step (1.) above is also a polynomial function of the features x and that the parameters are set so that $\bar{r}_{ij} \in [0, 1]$ with high probability. In this experiment, the number of features p is again kept fixed at 5. We varied the training set size $n \in \{100, 1000\}$, the parameter $\deg \in \{1, 4, 8, 16\}$, and the noise level parameter $\tau \in \{1, 2\}$. For each combination of these two parameters, we again ran 50 independent trials and used a test set of size 10,000. We do not use any regularization in this experiment, we use the SGD approach for training the SPO+ model of method (1.), and in the same manner as before we directly solve the optimization problems of methods (2.) and (3.) using Gurobi. Note that, out of fairness to the other methods, we did not use the early stopping technique for SGD.

Figure 3 displays our results for this experiment. Generally we observe similar patterns as in the shortest path experiment, although comparatively larger values of deg are needed to demonstrate the relative superiority of SPO+. In summary across all of our experiments, our results indicate that as long as there is some degree of model misspecification, then SPO+ tends to offer significant value over competing approaches, and this value is further strengthened in cases where more data available. The SPO+ approach is either always close to the best approach, or dominating all other approaches, making it a fairly suitable choice across all parameter regimes.

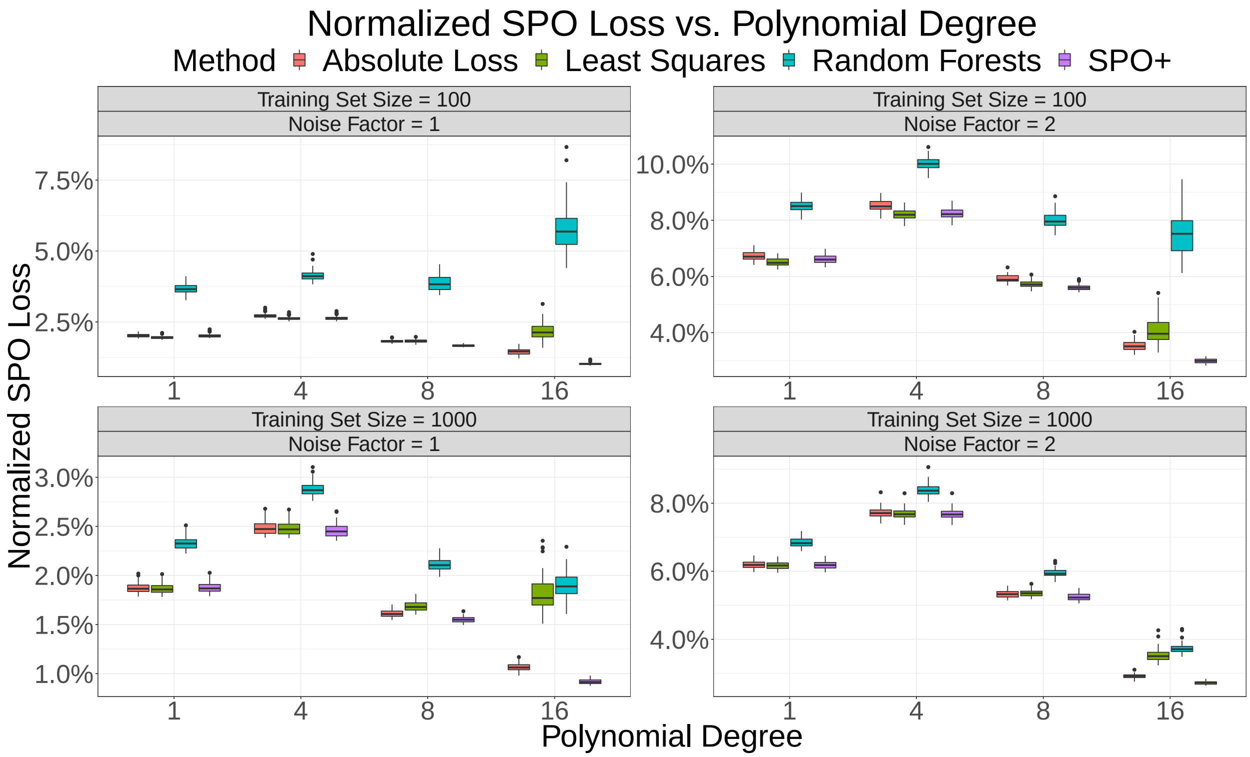


Figure 3 **Normalized test set SPO loss for the SPO+, least squares, absolute loss, and random forests methods on portfolio optimization instances.**

7. Conclusion

In this paper, we provide a new framework for developing prediction models under the predict-then-optimize paradigm. Our SPO framework relies on new types of loss functions that explicitly incorporate the problem structure of the optimization problem of interest. Our framework applies for any problem with a convex feasible region and a linear objective, even when there are integer constraints.

Since the SPO loss function is nonconvex, we also derived the convex SPO+ loss function using several logical steps based on duality theory. Moreover, we prove that the SPO+ loss is consistent with respect to the SPO loss, which is a fundamental property of any loss function. In fact, our results also directly imply that the least squares loss function is also consistent with respect to the SPO loss. Thus, least squares performs well when the ground truth is near linear, although, at least empirically, SPO+ strongly outperforms all approaches when there is a significant degree of model misspecification. Naturally, there are many important directions to consider for future work including more empirical testing and case studies, handling unknown parameters in the constraints, and dealing with nonlinear objectives.

Acknowledgements

The authors gratefully acknowledge the support of NSF Awards CMMI-1763000, CCF-1755705, and CMMI-1762744.

References

Ahuja, Ravindra K, Thomas L Magnanti, James B Orlin. 1993. Network flows: theory, algorithms, and applications .

Angalakudati, Mallik, Siddharth Balwani, Jorge Calzada, Bikram Chatterjee, Georgia Perakis, Nicolas Raad, Joline Uichanco. 2014. Business analytics for flexible resource allocation under random emergencies. *Management Science* **60**(6) 1552–1573.

Aswani, Anil, Zuo-Jun Max Shen, Auyon Siddiq. 2015. Inverse optimization with noisy data. *arXiv preprint arXiv:1507.03266* .

Balkanski, Eric, Aviad Rubinstein, Yaron Singer. 2016. The power of optimization from samples. *Advances in Neural Information Processing Systems*. 4017–4025.

Balkanski, Eric, Aviad Rubinstein, Yaron Singer. 2017. The limitations of optimization from samples. *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 1016–1027.

Ban, Gah-Yi, Noureddine El Karoui, Andrew EB Lim. 2016. Machine learning and portfolio optimization. *Management Science* .

Ben-David, Shai, Nadav Eiron, Philip M Long. 2003. On the difficulty of approximately maximizing agreements. *Journal of Computer and System Sciences* **66**(3) 496–514.

Bertsekas, Dimitri P. 1973. Stochastic optimization problems with nondifferentiable cost functionals. *Journal of Optimization Theory and Applications* **12**(2) 218–231.

Bertsekas, Dimitri P. 1999. *Nonlinear programming*. Athena scientific Belmont.

Bertsimas, Dimitris, Vishal Gupta, Nathan Kallus. 2013. Data-driven robust optimization. *Mathematical Programming* 1–58.

Bertsimas, Dimitris, Vishal Gupta, Nathan Kallus. 2014. Robust sample average approximation. *arXiv preprint arXiv:1408.4445* .

Bertsimas, Dimitris, Vishal Gupta, Ioannis Ch Paschalidis. 2015. Data-driven estimation in equilibrium using inverse optimization. *Mathematical Programming* **153**(2) 595–633.

Bertsimas, Dimitris, Nathan Kallus. 2014. From predictive to prescriptive analytics. *arXiv preprint arXiv:1402.5481* .

Bertsimas, Dimitris, Aurélie Thiele. 2006. Robust and data-driven optimization: modern decision making under uncertainty. *Models, Methods, and Applications for Innovative Decision Making*. INFORMS, 95–122.

Besbes, Omar, Yonatan Gur, Assaf Zeevi. 2015. Optimization in online content recommendation services: Beyond click-through rates. *Manufacturing & Service Operations Management* **18**(1) 15–33.

Borwein, Jonathan, Adrian S Lewis. 2010. *Convex analysis and nonlinear optimization: theory and examples*. Springer Science & Business Media.

Bottou, Léon. 2012. Stochastic gradient descent tricks. *Neural networks: Tricks of the trade*. Springer, 421–436.

Bottou, Léon, Frank E Curtis, Jorge Nocedal. 2016. Optimization methods for large-scale machine learning. *arXiv preprint arXiv:1606.04838* .

Chan, Carri W, Vivek F Farias, Nicholas Bambos, Gabriel J Escobar. 2012. Optimizing intensive care unit discharge decisions with patient readmissions. *Operations research* **60**(6) 1323–1341.

Chan, Carri W, Linda V Green, Yina Lu, Nicole Leahy, Roger Yurt. 2013. Prioritizing burn-injured patients during a disaster. *Manufacturing & Service Operations Management* **15**(2) 170–190.

Chan, Timothy CY, Tim Craig, Taewoo Lee, Michael B Sharpe. 2014. Generalized inverse multiobjective optimization with application to cancer therapy. *Operations Research* **62**(3) 680–695.

Cheung, Maurice, Adam N Elmachtoub, Retsef Levi, David B Shmoys. 2016. The submodular joint replenishment problem. *Mathematical Programming* **158**(1-2) 207–233.

Chu, Leon Yang, J George Shanthikumar, Zuo-Jun Max Shen. 2008. Solving operational statistics via a bayesian analysis. *Operations Research Letters* **36**(1) 110–116.

Cohen, Maxime C, Ngai-Hang Zachary Leung, Kiran Panchamgam, Georgia Perakis, Anthony Smith. 2017. The impact of linear optimization on promotion planning. *Operations Research* **65**(2) 446–468.

Den Boer, Arnoud V, Dirk D Sierag. 2016. Decision-based model selection. *https://www.researchgate.net/publication/282075352_Decision-based_model_selection* .

den Hertog, Dick, Krzysztof Postek. 2016. Bridging the gap between predictive and prescriptive analytics-new optimization methodology needed. *http://www.optimization-online.org/DB_HTML/2016/12/5779.html* .

Deng, Yunxiao, Junyi Liu, Suvrajeet Sen. 2018. Coalescing data and decision sciences for analytics. *Recent Advances in Optimization and Modeling of Contemporary Problems*. INFORMS, 20–49.

Deo, Sarang, Kumar Rajaram, Sandeep Rath, Uday S Karmarkar, Matthew B Goetz. 2015. Planning for hiv screening, testing, and care at the veterans health administration. *Operations research* **63**(2) 287–304.

Donti, Priya L, Brandon Amos, J Zico Kolter. 2017. Task-based end-to-end model learning. *arXiv preprint arXiv:1703.04529* .

Drusvyatskiy, Dmitriy, Adrian S Lewis. 2011. Generic nondegeneracy in convex optimization. *Proceedings of the American Mathematical Society* 2519–2527.

Dunning, Iain, Joey Huchette, Miles Lubin. 2017. Jump: A modeling language for mathematical optimization. *SIAM Review* **59**(2) 295–320.

Esfahani, Peyman Mohajerin, Soroosh Shafieezadeh-Abadeh, Grani Adiwena Hanasusanto, Daniel Kuhn. 2015. Data-driven inverse optimization with incomplete information. *arXiv preprint arXiv:1512.05489* .

Ferreira, Kris Johnson, Bin Hong Alex Lee, David Simchi-Levi. 2015. Analytics for an online retailer: Demand forecasting and price optimization. *Manufacturing & Service Operations Management* **18**(1) 69–88.

Gallien, Jérémie, Adam J Mersereau, Andres Garro, Alberte Dapena Mora, Martín Nóvoa Vidal. 2015. Initial shipment decisions for new products at zara. *Operations Research* **63**(2) 269–286.

Goh, Chong Yang, Patrick Jaillet. 2016. Structured prediction by least squares estimated conditional risk minimization. *arXiv preprint arXiv:1611.07096* .

Hofmann, Thomas, Bernhard Schölkopf, Alexander J Smola. 2008. Kernel methods in machine learning. *The annals of statistics* 1171–1220.

Kao, Yi-hao, Benjamin V Roy, Xiang Yan. 2009. Directed regression. *Advances in Neural Information Processing Systems*. 889–897.

Keshavarz, Arezou, Yang Wang, Stephen Boyd. 2011. Imputing a convex objective function. *Intelligent Control (ISIC), 2011 IEEE International Symposium on*. IEEE, 613–619.

Kleywegt, Anton J, Alexander Shapiro, Tito Homem-de Mello. 2002. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization* **12**(2) 479–502.

Lacoste-Julien, Simon, Mark Schmidt, Francis Bach. 2012. A simpler approach to obtaining an o (1/t) convergence rate for the projected stochastic subgradient method. *arXiv preprint arXiv:1212.2002* .

Levi, Retsef, Robin O Roundy, David B Shmoys. 2006. Primal-dual algorithms for deterministic inventory problems. *Mathematics of Operations Research* **31**(2) 267–284.

Lim, Andrew EB, J George Shanthikumar, Gah-Yi Vahn. 2012. Robust portfolio choice with learning in the framework of regret: Single-period case. *Management Science* **58**(9) 1732–1746.

Liyanage, Liwan H, J George Shanthikumar. 2005. A practical inventory control policy using operational statistics. *Operations Research Letters* **33**(4) 341–348.

Markowitz, Harry. 1952. Portfolio selection. *The journal of finance* **7**(1) 77–91.

Mehrotra, Mili, Milind Dawande, Srinagesh Gavirneni, Mehmet Demirci, Sridhar Tayur. 2011. Or practice—production planning with patterns: A problem from processed food manufacturing. *Operations Research* **59**(2) 267–282.

Mišić, Velibor, Georgia Perakis. 2019. Data analytics in operations management: A review. *Manufacturing & Services Operations Management, Forthcoming* .

Molchanov, Ilya. 2005. *Theory of random sets*, vol. 19. Springer.

Nemirovski, Arkadi, Anatoli Juditsky, Guanghui Lan, Alexander Shapiro. 2009. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization* **19**(4) 1574–1609.

Osokin, Anton, Francis Bach, Simon Lacoste-Julien. 2017. On structured prediction theory with calibrated convex surrogate losses. *arXiv preprint arXiv:1703.02403* .

Robbins, H., S. Munro. 1951. A stochastic approximation method. *Annals of Mathematical Statistics* **22** 400–407.

Rudin, Cynthia, Gah-Yi Vahn. 2014. The big data newsvendor: Practical insights from machine learning .

Schütz, Peter, Asgeir Tomasgard, Shabbir Ahmed. 2009. Supply chain design under uncertainty using sample average approximation and dual decomposition. *European Journal of Operational Research* **199**(2) 409–419.

Sen, Suvrajeet, Yunxiao Deng. 2017. Learning enabled optimization: Towards a fusion of statistical learning and stochastic optimization. *http://www.optimization-online.org/DB_HTML/2017/03/5904.html* .

Simchi-Levi, David. 2013. Om forum—om research: From problem-driven to data-driven research. *Manufacturing & Service Operations Management* **16**(1) 2–10.

Steinwart, Ingo. 2002. Support vector machines are universally consistent. *Journal of Complexity* **18**(3) 768–791.

Strassen, Volker. 1965. The existence of probability measures with given marginals. *The Annals of Mathematical Statistics* 423–439.

Tulabandhula, Theja, Cynthia Rudin. 2013. Machine learning with operational costs. *Journal of Machine Learning Research* **14**(1) 1989–2028.

Wagner, Harvey M, Thomson M Whitin. 1958. Dynamic version of the economic lot size model. *Management science* **5**(1) 89–96.

Wang, Zizhuo, Peter W Glynn, Yinyu Ye. 2016. Likelihood robust optimization for data-driven problems. *Computational Management Science* **13**(2) 241–261.