Learning Enabled Optimization: Towards a Fusion of Statistical Learning and Stochastic Programming

Yunxiao Deng, Suvrajeet Sen
Daniel J. Epstein Department of Industrial and Systems Engineering
University of Southern California, Los Angeles, 90089
yunxiaod@usc.edu, s.sen@usc.edu

Several emerging applications call for a fusion of statistical learning (SL) and stochastic programming (SP). The Learning Enabled Optimization paradigm fuses concepts from these disciplines in a manner which not only enriches both SL and SP, but also provides a framework which supports rapid model updates and optimization, together with a methodology for rapid model-validation, assessment, and selection. Moreover, in many "big data/big decisions" applications, these steps are repetitive, and realizable only through a continuous cycle involving data analysis, optimization, and validation. This paper sets forth the foundation for such a framework by introducing several novel concepts such as *statistical optimality, hypothesis tests for model-fidelity, generalization error of stochastic programming,* and finally, a *non-parametric methodology for model selection.* These new concepts provide a formal framework for modeling, solving, validating, and reporting solutions for Learning Enabled Optimization (LEO). We illustrate the LEO framework by applying it to an inventory control model in which we use demand data available for ARIMA modeling in the statistical package "R". In addition, we also study a production-marketing coordination model based on combining a pedagogical production planning model with an advertising data set intended for sales prediction.

Key words: Stochastic Linear Programming, Statistical Learning, Model Assessment

10/1/2018

1. Introduction

In recent years, optimization algorithms have become the work-horse of statistical (or machine) learning. Whether studying classification using linear/quadratic programming for support vector machines (SVM) or logistic regression using a specialized version of Newton's method, deterministic optimization algorithms have provided a strong foundation for statistical learning (SL)[1]. Indeed, SL could be labeled as "optimization enabled learning". The class of models studied in this paper, entitled Learning Enabled Optimization (LEO),

[1] We will use the term statistical learning when the discussion is application-agnostic. When referring to specific applications we use the term machine learning.

is intended to leverage advances in statistical learning to support the work-flow associated with stochastic programming (SP).

In the realm of SP, it is customary to model uncertainty via random variables, which essentially means that there exists a reasonable hope of approximating a distribution function to model random variables. Traditionally, the general statement of SP (Birge and Louveaux (2011)) is given in the form of the following optimization problem

$$\min_{x\in X} c(x) + \mathbb{E}[H(x,\tilde{\omega})], \tag{1}$$

where x and X denote a vector of decision variables and a feasible set (resp.), c denotes a lower-semicontinous cost function, $\tilde{\omega}$ is a given random variable (with known distribution), which induces the random variable H whose outcomes h provide the so-called recourse (function) value (i.e., value function of another optimization model). In cases where the $\tilde{\omega}$ obeys a continuous distribution or even an empirical distribution with a large number of data points, one uses sampling to setup a more manageable version known as the Sample Average Approximation (SAA) problem. In the SL literature, such an approximation is known as Empirical Risk Minimization.

In order to pose the kind of questions which motivate this paper, consider a common SL situation in which we have a dataset of i.i.d. observations (Z_i, W_i) which represent covariates (Z, W), generically referred to as predictors and responses respectively. Unlike SP, where the distribution is assumed to be available, the goal of a learning model is to discover a relationship (e.g., a regression) between the covariates (Z, W). These statistical relationships can reveal a treasure-trove of information. Depending on the model, these relationships can predict outcomes of the response variable, given a realization $Z = z$. Thus a LEO model is a parametric version of an SP stated as follows.

$$\min_{x\in X} c(x) + \mathbb{E}[H(x, W|Z = z)], \tag{2}$$

where $\mathbb{E}[H(x, W|Z = z)]$ is the conditional expectation, given the parameter z. Note however, that unlike SP, the distribution of the random variable $W|Z = z$, is not available directly. Instead we may infer a distribution of $W|Z = z$ using the available dataset. Moreover, since we will be working with finite datasets, we will adopt the SL orientation based on training and validation, which requires that we undertake statistical tests a posteriori (after optimization).

In some applications, the predictors Z assume values which do not belong to the same space as the decisions x. In such cases, the resulting model is said to be "LEO with disjoint spaces." At the other end of the spectrum, if $z = x$ in (2), then the resulting model is said to be "LEO with shared spaces." Such models often arise in areas such as design optimization, marketing optimization, and similar circumstances where decisions can be interpreted as "bets". The analogous SP models are known as problems with decision-dependent uncertainty.The mathematical structures for these two types of LEO models may be quite different (e.g., convexity or lack thereof).

Because the emphasis of SL is to work directly with the dataset (instead of a distribution), our plan for fusion involves using regression models to infer stochastic models, which can be used to represent the uncertainty impacting the decision environment. The current paper adopts a two-stage stochastic programming approach, and we address several open questions as summarized below.

1. (Modeling) Expected Risk Minimization in SL and Sample Average Approximation in SP are subsumed under (1). However when data is available as covariate (Z, W), then how should we accommodate decision models such as (2), where the conditional expectation can only be approximated? The process of creating such approximations will be referred to as Learning, and the methodology for solving the decision problem is referred to as Optimization.

2. (Statistical Optimality) In the presence of stochastic forecasts from SL, what statistical estimates can we provide regarding the probability of δ-optimality of a proposed decision $(\delta > 0)$?

3. (Generalizability) How should one estimate "generalizability" which provides cost-to-go estimates of a decision beyond "in-sample" data (i.e., the ability to perform reasonably well even for outcomes which are not included within the training set)? Can we provide other commonly used statistical metrics (e.g., 95% prediction interval of costs) to support decision-making.

4. (Model Validation and Selection) How should we validate the effectiveness of data-driven models so that we can defend decisions as being validated via statistical principles? Moreover, since it is common to examine multiple SL models, how should one select a model which provides the most appropriate decision?

This paper is organized as follows. Connections to the literature are presented in section 2. In section 3, we present the fundamental structures, namely,"LEO Models with Disjoint Spaces" and "LEO Models with Shared Spaces". We illustrate the first of these structures with an inventory control problem, and the second one is illustrated via a production-marketing coordination problem. Because LEO models will allow both continuous and discrete random variables, the statement of optimization will be relaxed to solutions with high probability (greater than 95%, say) of optimality. This concept, which is set forth in section 4, will be referred to as "statistical optimality" for sequential sampling algorithms. In section 5 we study hypothesis tests for model validation. Such tests identify the contenders (models) which may be most promising. In addition, we also define a concept of generalization error for optimization. For LEO models, this measure aims to quantify the degree of flexibility expected in the decision model. This entire protocol is illustrated in section 6 via computations for the examples introduced in section 3. Finally, section 7 presents our conclusions and possible paths forward.

2. Connections to the Literature

In terms of scientific genealogy, one can trace the introduction of learning into optimization from the work on approximate dynamic programming (ADP, Bertsekas (2012), Powell (2011)) and approximate linear programming (ALP, e.g. De Farias and Van Roy (2004)). The canonical structure of these approaches pertains to DP, where one uses approximations of the DP value function by using basis functions. In this paper, the canonical setup derives from constrained optimization, although we will state our objectives in the context of approximate solutions. In this sense, one refers to the technical content of our approach as "approximate stochastic programming."

An alternative data-driven approach to modeling uncertainty is via the ideas of Robust Optimization (RO) (Bertsimas and Sim (2004), Ben-Tal and Nemirovski (2001)), where performance is measured in terms of the ability to withstand extreme or near-extreme events. There are many applications (e.g., engineering design) where the ability to survive unforeseen circumstances is important. Slightly less demanding criteria come about via risk-averse optimization where the decision-making model attempts to strike a balance between "risk" and "return"(e.g., Miller and Ruszczyński (2011)). The line of work pursued in this paper pertains to SP using covariate data and estimated conditional expectation as in (2). With regard to data-driven SP, we mention the recent work of Van Parys et al. (2017)[2] where it is shown that minimizing disappointment (or regret) within a data-driven SP automatically leads to the solution of a Distributionally Robust Optimization (DRO) model. Nevertheless, their paper focuses on a data-driven version of (1), not (2). All of the above models serve alternative types of applications of decisions under uncertainty, and they each possess their "sweet spot" in terms of applications and ability to cope with constrained decisions under uncertainty.

In keeping with our goals to accomplish more with data in OR/MS applications, there have been some studies of optimization methods with information gathering in predictive analytics (Frazier (2012) and Ryzhov et al. (2012)). That line of work is intended to help experimentalists improve the effectiveness of predictive models by using sensitivity of the response, using a concept known as knowledge gradient. Their work uses algorithmic notions of optimization for experimental design (including simulation experiments). A more decision-oriented framework is proposed in Kao et al. (2009) where the regression coefficients are chosen based on a combined objective consisting of a loss function associated with regression, and a quadratic objective for optimization. The assumptions of Kao et al. (2009), namely, unconstrained quadratic optimization of both problems renders their simultaneous optimization manageable. However, if the optimization model were inequality constrained (as in many applications), such simultaneous optimization would lead to bilevel stochastic programs, which are much harder than the SP setting of the LEO model. Another viewpoint at the interface between predictive and prescriptive analytics is presented in the recent paper by Bertsimas and Kallus (2014). Their work demonstrates that in presence of random covariates (as in (2)), direct usage of SAA (using the data set $\{Z_i, W_i\}$) can produce decisions which are not consistent with optimality, even when the size of the dataset grows indefinitely. The reasons underlying this phenomenon may be attributed to the difference in model representations in (1) and (2).

Before closing this section, we also mention that the OM literature has adopted a combined learning and optimization focus on specific classes of problems. For instance, Liyanage and Shanthikumar (2005) and more recently Ban and Rudin (2017) have studied the integration of optimization and learning to identify optimal inventory ordering policies.

[2] This paper appears to be have been completed around the same time as the first submission of our paper

Both papers use the specialized form of the newsvendor model to illustrate the potential for learning and optimization. Another avenue of application-specific use of learning within optimization arises from the introduction of specialized regularization in the context of portfolio optimization in Ban et al. (2017). While these special cases have made a case for the potential for the fusion of SL and SP, this paper focuses on some general principles of fusion, via LEO.

3. Learning Enabled Optimization

Statistical Learning provides support for predictive analytics, whereas, optimization forms the basis for prescriptive analytics, and the methodologies for these are built somewhat independently of each other. The process recommended for SL is summarized in Figure 1a in which the entire data set is divided into two parts (Training and Validation), with the former being used to learn model parameters, and the latter dataset is used for model assessment and selection. Once a model is selected, it can be finally tested via either simulation or using an additional “test dataset” for trials before adoption. This last phase is not depicted in Figure 1 because the concepts for that the test phase can mimic those for the model validation phase. Finally, note that in Figure 1b, we insert the prescriptive model to allow the LEO framework to adopt alternative paths for a fusion of Predictive and Prescriptive models.

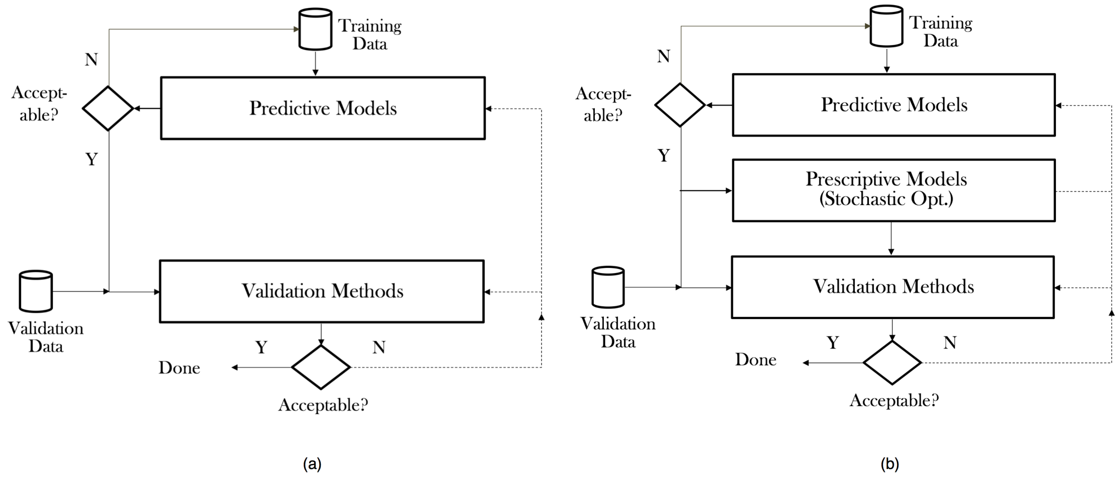

Figure 1 Statistical Learning and Learning Enabled Optimization

3.1. Modeling Protocol

This subsection presents our aspirations for LEO models. As one might expect, this framework consists of two major pieces: the SL piece and the SP piece. We begin by stating a regression model in its relatively standard form. Towards this end, let m denote an arbitrary regression model using a training dataset (T). For notational simplicity we assume that $W_i \in \mathbb{R}$, whereas $Z_i \in \mathbb{R}^p$. Given the training data, a class of deterministic models $\hat{\mathcal{M}}$, and a loss function L, a regression may be represented as follows:

$$\hat{m} \in \operatorname{argmin}\Big\{\frac{1}{|T|}\sum_{i\in T} L(W_i, m(Z_i)) \mid m \in \hat{\mathcal{M}}\Big\}. \tag{3}$$

We emphasize that in many circumstances (e.g., modeling the impact of natural phenomena such as earthquakes, hurricanes etc.), model fidelity may be enhanced by building the statistical model of the phenomena, independently of decision models. For such applications, one may prefer to treat the development of the SL piece prior to instantiating the SP piece. In other cases, the models may have greater symbiosis with tightly coupled interactions. Thus a coupled LEO model is possible due to the undirected link into the prescriptive box of Figure 1(b), thus allowing the prescriptive model to relay decision data to the predictive model if necessary.

3.2. Alternative Error Models

Assumption 1 (A1). $\{Z_i, W_i\}$ are assumed to be i.i.d. observations of the data proces. Moreover we assume that the errors are homoscedastic. This is a standard assumption for regression models.

Assumption 2 (A2). We will assume that decisions in the SP model, denoted x, have no impact on the continuing data process $\{(Z,W)\}$ to be observed in the future.

To put this assumption in the context of some applications, note that in the advertising/financial market, it may be assumed that an individual advertiser/investor is not large enough to change future market conditions.

Empirical additive errors: Suppose that an error random variable has outcomes defined by $\xi_i := W_i - \hat{m}(Z_i)$ as in (3). By associating equal weights to these outcomes, one obtains a discrete random variable with an empirical distribution. For the case of multiple linear regression (MLR), $\hat{m}(z) = \sum_\tau \beta_\tau z_\tau$, where τ ranges over the index set of predictors. In that case, the empirical additive errors are given by substituting the deterministic affine function in place of $\hat{m}$. For $\tau = 0$, it is customary to use $z_\tau = 1$. When the distribution of the error term does not depend on any specific choice of $Z = z$, the random variable is said to be homoscedastic (as is commonly assumed for MLR). With this assumption, the data for the error terms ξ become applicable to any choice of z. Therefore $\hat{m}(z) + \xi$ becomes the representation of the random variable $W|Z = z$ in (2). While on the context of error distributions, we should mention the work of Rios et al. (2015) which uses splines and epi-splines to model additive error distributions (see also Royset and Wets (2014)). However, those error distributions may not satisfy homoscedasticity, and in that case, the resulting models may be more difficult, and extensions to non-convex optimization would be required.

For a more general setup (known as projection pursuit (Friedman and Stuetzle (1981))), one may define $\hat{m}(z) = \sum_{\tau \in \mathcal{T}} \phi_\tau(\beta_\tau^\top z)$, where $\mathcal{T}$ is a finite index set. Again, the errors are assumed to have the same homoscedasticity properties. We should note that the notion of homoscedasticity translates to stationary error distributions in the context of time series models as well, and there are standard (and partial) autocorrelation tests which can discern stationarity.

Multi-dimensional errors: It has been recognized that using fixed regression coefficients can give rise to overfitting in SL models. Define a stochastic function $m(z, \xi) = \sum_\tau \tilde{\beta}_\tau \phi_\tau(z)$, where $\phi_0 = 1$, and $\phi_\tau(\cdot)$, are deterministic functions, but the parameters $\tilde{\beta}_\tau$ are elements of a multivariate random variable. Let $\hat{m}(z) = \sum_\tau \bar{\beta}_\tau \phi_\tau(z)$, where $\bar{\beta}_\tau = \mathbb{E}(\beta_\tau)$. In this case, a vector of errors associated with an outcome of random coefficients $\{\tilde{\beta}_\tau\}$ is given by the difference $\tilde{\xi}_\tau = \tilde{\beta}_\tau - \mathbb{E}(\beta_\tau)$. The coefficients $\{\tilde{\beta}_\tau\}$ may be correlated, but the multivariate distribution of these coefficients $\{\tilde{\beta}_\tau\}$ are assumed to not depend on any specific choice of $Z = z$ (homoscedasticity). With this assumption, we can represent the random variable $W|Z$ in (2) by $m(z, \xi) = \sum_\tau \tilde{\beta}_\tau \phi_\tau(z)$. Such random coefficients are also being considered for neural networks to improve their flexibility (see Blundell et al. (2015) and Pawlowski et al. (2017)).

Remark 1. (a) In SL it is common to formulate a stochastic model $m(z, \xi)$, although predictions are deterministic and made with $\hat{m}(Z = z)$. In contrast, LEO will use an optimization model which will recognize inherent randomness of $m(x, \xi)$. For this reason, the deterministic model $\hat{m}(z)$ is not as relevant (for LEO) as the stochastic model $m(z, \xi)$. Such stochasticity will provide the optimization model a sense of uncertainty faced by the decision-maker, thus reducing overfitting (i.e., improving generalizability). Nevertheless, the decisions produced by LEO must also be deterministic (at least for the first period, i.e., here-and-now). This feature of allowing randomness in SL also distinguishes LEO from Directed Regression, which uses the deterministic model $\hat{m}(z)$ rather than the stochastic model $m(z,\xi)$. (b) Note that alternative stochastic models m may lead to the same deterministic model $\hat{m}$. This is certainly the case with MLR. For this reason, the LEO setting will allow a finite list of plausible alternative stochastic models which will be indexed by the letter q. We suggest representing the class of models by $(\ell_q, \hat{\mathcal{M}}_q)$, and the specific deterministic model obtained by (3) is denoted $\hat{m}_q$. Whenever the specific index of a model is not important, we will refer to deterministic model as $\hat{m}$, and its stochastic counterpart by m. ■

For the remainder of this section, we present the two alternative structures for the SP part of a LEO model.

3.3. LEO Models with Disjoint Spaces.

Let $x \in \mathbf{X} \subseteq \mathbb{R}^{n_1}$ denote the optimization variables, and suppose that the predictors Z have p elements indexed by the set $\mathcal{J} = \{1,...,p\}$. The simplest version of a LEO model is one in which the predictors Z do not assume values in the space of optimization variables (x). As suggested in Remark 1(b), SL models m_q are now assumed to be given. For the case of MLR, the parameters of the regression are random variables with a corresponding distribution $\mathcal{P}_q$. Then we have an objective function of the following form.

$$f_q(x) := c(x) + \mathbb{E}_{\xi_q}[H(x, \tilde{\xi}_q | Z = z)]. \tag{4}$$

Note that the (4) has the form similar to (2) except the random variable is defined by an error term $\tilde{\xi}_q | Z$, which induces a random variable H whose outcomes h are given by

$$h(x, \xi_q | Z = z) = \min\{d(y) | g(x,y) - m_q(z, \xi_q) \leq 0, y \in \mathbf{Y} \subseteq \mathbb{R}^{n_2}\}, \tag{5}$$

with $g: \mathbb{R}^{n_1+n_2} \to \mathbb{R}$. Clearly these constraints could be multi-dimensional, but some of the same conceptual challenges would persist. Nevertheless, there is an important algorithmic advantage resulting from the disjoint structure: because z and x belong to disjoint spaces, optimization with respect to x is not restricted by z. Hence, (4) can optimized by simply passing predicted values $m_q(z, \xi_q)$. As a result, the complexity of m_q does not impact the optimization part of the LEO model, and very general regressions (e.g., Kernel-regression and others) can be included for the prediction process in (4).

Assumption 3 (A3-a). We assume that c, d, g are convex functions, the sets $\mathbf{X}, \mathbf{Y}$ are convex sets, and $h(x, \cdot)$ is a convex function in x as well.

The value function h defined in (5) goes by several alternative names such as the "recourse" function in SP or "cost-to-go" function in DP. While the underpinnings of LEO models are closer to SP than DP, we adopt the "cost-to-go" terminology because we plan to extend LEO models to allow other types of "cost" predictions in future papers (e.g., forecasts of computational times, and other elements of an uncertain future).

3.4. LEO Models with Shared Spaces.

We continue with assumptions A1- A3, and will include another assumption for this class of models. Consider an SP model in which there exists a non-empty subset $J \subseteq (\{1, \ldots, p\} \cap \{1, \ldots, n_1\})$ such that the decision space $\mathbf{X}$ and the sample space $\mathcal{Z}$ share the same subspace, i.e., $\mathbf{X}_J = \mathcal{Z}_J$, where the subscript represents the projection onto the subspace indexed by J. Hence these models will be referred to as "models with Shared Spaces", and the objective function is

$$f_q(x) := c(x) + \mathbb{E}_{\xi_q}[H(x, \tilde{\xi}_q | Z = z, z_r = x_r, r \in J)]\}, \tag{6}$$

where, H is a random convex function over the feasible set $\mathbf{X}$, and the outcomes h are defined as follows.

$$h(x, \xi_q | Z = z, z_j = x_j, j \in J) := \min\{d(y) | g(x, y) - m_q(z, \xi_q) \leq 0, y \in \mathbf{Y} \subseteq \mathbb{R}^{n_2}\} \tag{7}$$

As in (5) we assume that $h(x, \cdot)$ defined in (7) is a convex function in x. Note that in this form of the model, the decision maker is called upon to make a "bet" $(x_j, j \in J)$, and the response is a random variable $H(x, \tilde{\xi}_q | Z = z, z_j = x_j, j \in J)$. While, the objective function of both Disjoint and Shared Spaces models have a similar forms, the interplay between decisions and random variables are different. To accommodate this, we state the following assumption.

Assumption 3 (A3-b). In addition to Assumption (A3-a), (7) imposes the assumption that $m_q(z, \xi_q)$ is concave in $z_j, j \in J$ for all ξ except on a set of $\mathcal{P}_q$-measure zero.

Proposition 1. (a) Under assumptions A1 and A2, the objective function of problem (6) provides a consistent estimator of the optimal cost-to-go value as the number of data points approaches infinity.

(b) Under assumption A3, problem (6) is convex, and each instance in (7) is also convex.

Proof: Part (a) follows from section 2.4 of Hastie et al. (2011). Part (b) is a straightforward application of the convexity assumptions.

3.5. Statistical Aspiration of a LEO Model.

The mathematical statement of a LEO model will be based on the recognition that importing a statistical model into an optimization problem can be demanding, especially when the number of predictors is large. However, the payoff (via increased flexibility and greater optimality) may be significant. Due to the possibility that the parameters of a statistical model (e.g. MLR) can have several coefficients which are continuous random variables (e.g, Gaussian), evaluating $f(x)$ may require some sampling-based SP algorithm. One of the more popular approaches for such SP is the Sample Average Approximation (SAA) which is summarized in Appendix II. However, it has been observed by many authors (e.g. Homem-de Mello and Bayraksan (2014)) that the sample size recommended by SAA theory can be extremely conservative. For this reason, the algorithmic use of SAA consists of solving a sequence of sampled problems, each using a larger sample size (e.g., Linderoth et al. (2006)). It is fair to suggest that seeking deterministic optimality is far too demanding for the case of large number of predictors in the case of LEO models with shared spaces.

In order to bring the aspirations of a modeler (in search of decisions with performance guarantees), we state the LEO model in terms of a probabilistic guarantee. Let $\mathcal{P}_q$ denote the probability distribution of a random variable ξ_q. For LEO models, we will seek a pair (x_q, γ_q) such that for a pre-specified accuracy tolerance $\delta > 0$, we have

$$\gamma_q := \mathcal{P}_q \ (x_q \in \delta - \arg\min\{f_q(x) | x \in \mathbf{X}\}), \tag{8}$$

with $\gamma_q \geq \underline{\gamma}$ (= 0.95, say). As the reader might notice, (8) states our aspirations for a LEO model in such a manner that we can report the following critical quantities: δ, γ_q, x_q for a model indexed by q. The manner in which we verify these conditions will be discussed in the section on Statistical Optimality.

3.6. Examples of LEO Models

Due to the structural differences between (4) and (6) we will use a time-series model (ARIMA) to illustrate models with a disjoint structure, whereas, the illustration for the latter will be restricted to MLR because the model with shared spaces remains computationally tractable.

Example 1. Inventory Control: A LEO model with Disjoint Spaces -`LEO-ELECEQUIP`

The `ELECEQUIP` dataset in R provides 10 years of demand data for electrical equipment. We present an inventory control model with this data-set. Consider making equipment ordering choices in period t based on demand data from previous periods (i.e., periods $j < t$). Since the optimization model chooses decisions for periods $j \geq t$, we treat the optimization variables and that of the statistical model as disjoint. Clearly, this property holds for rolling horizon models as well. Because of the disjoint spaces structure, such a LEO model can entertain reasonably complex descriptions of data (e.g. time series, nonlinear and the so-called link functions). Our preliminary results provide an example using an ARIMA model, with (p,d,q), (P,D,Q) = (0,0,0), (1,1,0). This notation (p,d,q) is standard for ARIMA, and represents the number of periods used in predicting three parts of the stationary series (i.e., after removal of trends) representing autoregressive terms (p), integration terms (d) and moving average errors (q). The quantities P,D,Q refer to the "seasonal" component, which in this case was annual. In choosing these parameters, it is customary to test for stationarity using autocorrelation and partial autocorrelation functions (ACF and PACF) (Box et al. (2015)). Note that for this dataset P=1, D=1, the ARIMA model implies that two previous periods (from a year ago) are necessary for forecasting the demand for period $t = 1$. In order to acknowledge the rolling-horizon nature of the decision models, we also include an additional period look-ahead after period 1. Thus we have a three period model with $t = 0, 1, 2$. The detailed model formulation is provided in Appendix I. ■

Example 2. Production-Marketing Coordination: A LEO model with Shared Spaces - `LEO-Wyndor`

An important piece of data for production planning is predicted sales, which in turn depends on how much advertising is carried out. Suppose that a company buys advertising slots (units of time) on several media channels, then, this decision has an impact on future sales figures. We present an example which we refer to as the LEO-Wyndor data in which the decision vector x represents the allocation of the advertising time slots to each type of media (TV and radio). The name Wyndor and the production part of this problem is borrowed from a very popular OR textbook (Hillier and Lieberman (2012)). Our example extends the original Wyndor model to one in which the production plan is to be made while bearing in mind that the allocation of the advertising time slots affects sales of Wyndor products (two types of doors), and the final production plan will be guided by firm orders (sales) in the future. For this example, a statistical model predicting future sales is borrowed from the advertising dataset of James et al. (2013) which also presents an MLR model relating sales (W) with advertising slots (Z) on TV and radio. In this example, the advertising decisions constitute a "bet" on the first stage (advertising) decisions x, and the second stage decisions are the production planning choices, given "firm orders" (sales). A specific numerical instance of this model is given in Appendix I. ■

The models presented in Appendix I have the same structure as a stochastic linear program (SLP). In case of the `LEO-ELECEQUIP` example, the cost-to-go function h has the form $h(x,\xi) = \min\{d^\top y \mid Dy = \xi - Cx, y \geq 0\}$, leading to a SP with deterministic matrices C, D, while randomness only appears as a vector ξ on the right hand side of the second-stage model. This is also the structure studied in Bertsimas and Kallus (2014).

In case of the `LEO-Wyndor` example, several alternative LEO models are plausible based on the SL model used. When we use empirical additive errors (EAE), the MLR coefficients ($\beta_j, j = 0,1,2$) are fixed scalars, and they become the deterministic entries of the matrix C, while the empirical additive error ξ is the random right hand side. On the other hand, when the MLR model (with normally distributed coefficients) is used, then $\{\tilde{\beta}_j\}_{\{j=0,1,2\}}$, form entries of C which are also random. If cross-correlations among these coefficients are ignored, then the matrix C is diagonal (with normally distributed uncorrelated random variables - denoted NDU). On the other hand, if the cross-correlations of $\{\tilde{\beta}_j\}_{\{j=0,1,2\}}$ are included, then the resulting LEO model has error terms which are normally distributed and correlated (denoted NDC). While these alternative statistical models are plausible, statistical theory recommends that we choose the model with the least generalization error (in a statistical sense). Since the goal of LEO models is to recommend decisions which are both generalizable and statistically "near-optimum", the process involves computing appropriate metrics for both (see section 5).

4. Statistical Optimality

In this section, we assume that the index q is fixed, and hence it is suppressed below. The algebraic statements of most optimization models lead to algebraic optimality conditions, and in turn, those engender deterministic algorithms.In the context of SL, computational experience has suggested that seeking a truly deterministic optimum detracts from generalizability of a solution, which is also one of the reasons for the popularity of stochastic gradient descent. In this section we will introduce a quantifiable "reliability" requirement in (8) which prompts statistical optimality.

Statistical optimality bounds have been studied in the literature for a while (e.g., Higle and Sen (1991), Higle and Sen (1996b), Mak et al. (1999), Kleywegt et al. (2002), Bayraksan and Morton (2011), Glynn and Infanger (2013)). A complete mathematical treatment of these concepts appears under the banner of "Statistical Validation" in Shapiro et al. (2009), and a detailed tutorial for SAA appears in Homem-de Mello and Bayraksan (2014). In addition, there have been theoretical investigations on how the computational budget ought to be allocated so that increases in sample size can be determined in an online manner (e.g., Bayraksan and Pierre-Louis (2012), Royset and Szechtman (2013)). Despite this relatively long history, their use in identifying near-optimal *decisions* for realistic instances has been limited. There are at least two hurdles to overcome here: 1) as mentioned earlier, the sample size requirements predicted by the current theory leads to relatively large approximation problems, and 2) while replications of sampled algorithms are important for lower variance estimates of the objective function f, the unfortunate reality of sampling-based optimization is that replications may introduce significant variability in decisions (see experience with SSN reported in Freimer et al. (2012)). One remedy to overcome variability is to use compromise decisions as described in Appendix II. The current section extends the result of Appendix II to a situation in which statistical optimality of a compromise decision can be verified up to an accuracy of δ as in (8). To accomplish this goal, we combine the algorithmic framework of Sen and Liu (2016) and the convergence rate results of SAA (Chapter 5, Shapiro et al. (2009)) to obtain a distribution-free estimate of the probability of optimality of the proposed decision. Thus our approach combines concepts from external sampling (SAA), as well as internal sampling (SD) within one common framework. In the interest of preparing a self-contained presentation, we provide brief summaries of SAA and SD in the Appendix II[3].

In the following, we impose the assumptions required for the asymptotic convergence of SD. Let $\nu = 1, ..., M$ to denote the index of replications, and for each ν, the SD algorithm is assumed to run for $K_\nu(\varepsilon)$ samples, to produce a terminal solution $\mathbf{x}^\nu(\varepsilon)$, and a terminal value f^ν_ε, where ε is the stopping tolerance used for each replication. From Appendix II, note that the grand-mean approximation $\bar{F}_M(x) := (1/M)\sum_{\nu=1}^M f^\nu(x)$, where $\{f^\nu\}_{\nu=1}^M$ denotes terminal value function approximations for each replication m. In addition, $\bar{\mathbf{x}} = (1/M)\sum_\nu x^\nu$, and the compromise decision $\mathbf{x}^c$ is defined by $\mathbf{x}^c \in \arg\min\{\bar{F}_M(x) + \frac{\bar{\rho}}{2}||x - \bar{\mathbf{x}}||^2 : x \in \mathbf{X}\}$, where $\bar{\rho}$ is the sample average of $\{\rho^\nu\}$, which denote the terminal proximal parameter for the ν^{th} replication.

Theorem 1. Assume $\mathbf{X}$ is non-empty, closed and convex, and the approximations f^ν are proper convex functions over $\mathbf{X}$. For $\delta = \bar{\rho}||\mathbf{x}^c - \bar{\mathbf{x}}||^2$, we have,

$$\frac{1}{M}\sum_{\nu=1}^{M} f^\nu_\varepsilon + \delta \geq \bar{F}_M(\mathbf{x}^c). \tag{9}$$

which implies that $\mathbf{x}^c$ is δ-argmin of $\frac{1}{M}\sum_{\nu=1}^M f^\nu(\cdot)$, and the tolerance level satisfies $\delta = \bar{\rho}||\mathbf{x}^c - \bar{\mathbf{x}}||^2$.

Proof: Since $\mathbf{x}^c \in \arg\min\{\bar{F}_M(x) + \frac{\bar{\rho}}{2}||x - \bar{\mathbf{x}}||^2 : x \in \mathbf{X}\}$, we have,

$$0 \in \partial\bar{F}_M(\mathbf{x}^c) + \mathcal{N}_X(\mathbf{x}^c) + \bar{\rho}(\mathbf{x^c} - \bar{\mathbf{x}}).$$

Hence, $-\bar{\rho}(\mathbf{x^c} - \bar{\mathbf{x}})$ can be used as a subgradient of the function $\bar{F}_M(x) + \mathcal{I}_X(x)$ at $x = \mathbf{x}^c$. For all $x \in \mathbf{X}$,

$$\bar{F}_M(x) + \mathcal{I}_X(x) \geq \bar{F}_M(\mathbf{x}^c) + \mathcal{I}_X(\mathbf{x}^c) - \bar{\rho}(\mathbf{x}^c - \bar{\mathbf{x}})^\top(x - \mathbf{x}^c)$$

Since $\bar{\mathbf{x}}, \mathbf{x^c} \in \mathbf{X}$, the indicator terms vanish, and therefore,

$$\bar{F}_M(\bar{\mathbf{x}}) + \bar{\rho}(\mathbf{x}^c - \bar{\mathbf{x}})^\top(\bar{\mathbf{x}} - \mathbf{x}^c) \geq \bar{F}_M(\mathbf{x}^c).$$

Since $\bar{\rho}(\mathbf{x^c} - \bar{\mathbf{x}})^\top(\bar{\mathbf{x}} - \mathbf{x^c}) \leq \bar{\rho}||\mathbf{x^c} - \bar{\mathbf{x}}||\ ||\bar{\mathbf{x}} - \mathbf{x^c}||$, we have

$$\bar{F}_M(\bar{\mathbf{x}}) + \bar{\rho}||\mathbf{x^c} - \bar{\mathbf{x}}||^2 \geq \bar{F}_M(\mathbf{x}^c). \tag{10}$$

[3] Readers unfamiliar with the context of Appendix II are well advised to familiarize themselves with it before reading the remainder of this section

Recall that $\bar{\mathbf{x}} = \frac{1}{M}\sum_\nu \mathbf{x}^\nu$, and $\bar{F}_M$ is convex, therefore, $F_M(\bar{\mathbf{x}}) \leq \frac{1}{M}\sum_\nu \bar{F}_M(\mathbf{x}^\nu)$. Because $f^j(\mathbf{x}^\nu) \leq f^\nu(\mathbf{x}^\nu)$ for all pairs (j,ν), and $f^\nu_\varepsilon = f^\nu(\mathbf{x}^\nu)$, we have

$$\bar{F}_M(\bar{\mathbf{x}}) \leq \frac{1}{M}\sum_\nu \bar{F}_M(\mathbf{x}^\nu) \leq \frac{1}{M}\sum_\nu f^\nu_\varepsilon. \tag{11}$$

Combining (10) and (11), we get

$$\frac{1}{M}\sum_\nu f^\nu_\varepsilon + \delta = \frac{1}{M}\sum_\nu f^\nu_\varepsilon + \bar{\rho}||\mathbf{x^c} - \bar{\mathbf{x}}||^2 \geq \bar{F}_M(\mathbf{x}^c). \quad \blacksquare$$

If we define $\hat{S}_M(\delta) = \{x \in \mathbf{X} \mid \bar{F}_M(x) \leq \frac{1}{M}\sum_\nu f^\nu(\mathbf{x}^\nu) + \delta\}$, f^* the optimal value, and $S(\delta_u) = \{x \in \mathbf{X} \mid \bar{F}_M(x) \leq f^* + \delta_u\}$, then Theorem 1 has proved that $\mathbf{x}^c \in \hat{S}_M(\delta)$. Note that $S(\delta_u)$ defines the solution set which is δ_u-optimal to the true optimal solution, and as a result, we should also analyze the relationship between $\mathbf{x}^c$ and $S(\delta_u)$. The choice of δ_u should be greater than the stopping tolerance ε.

Unless one restricts the SL model to only Empirical Additive Errors (EAE), it is difficult for a user to prescribe a sample size for a stochastic programming model as discussed earlier in the section. Hence we do not recommend this approach for cases where the SL coefficients are normally distributed. Instead, we use SD to suggest sample sizes, and discover the probability that a recommendation $\mathbf{x}^c \in S(\delta_u)$. This approach can also be used by other SAA based SP algorithms which produce value function approximations such as inexact bundle methods (e.g., Oliveira et al. (2011)), regularized decomposition (Ruszczyński (1986)) and others.

Theorem 2. Let $F(x,\tilde{\xi}) := c(x) + H(x,\tilde{\xi})$ denote the objective functional random variable in either (4) or (6). Suppose for each outcome ξ, $\kappa(\xi)$ satisfies $|F(x',\xi) - F(x,\xi)| \leq \kappa(\xi)||x' - x||$. We define the Lipschitz constant of $\mathbb{E}_\xi[F(x,\tilde{\xi})]$ as $L = \mathbb{E}_\xi[\kappa(\tilde{\xi})]$. Suppose $\mathbf{X} \subseteq \mathbb{R}^n$ has a finite diameter D, M denotes the number of replications in solving the SP, while N denotes the minimum of sample size of all the replications, and let the tolerance level $\delta_u > \delta$, with δ defined in Theorem 1. Then we have the following inequality:

$$\Pr(\hat{S}_M(\delta) \subset S(\delta_u)) \geq 1 - \exp\left(-\frac{NM(\delta_u - \delta)^2}{32L^2D^2} + n\ln\left(\frac{8LD}{\delta_u - \delta}\right)\right). \tag{12}$$

Proof: If we solve for the probability from (24) in Proposition 2, the following inequality holds:

$$\Pr(\hat{S}_M(\delta) \subset S(\delta_u)) \geq 1 - \exp\left(-\frac{K(\delta_u - \delta)^2}{8\lambda^2 D^2} + n\ln\left(\frac{8LD}{\delta_u - \delta}\right)\right). \tag{13}$$

From assumption SAA-c in Appendix II, $\lambda = 2L$. Also, recall from (25) in Appendix II, each replication uses a sample size of at least N. Therefore, in this case the total sample size K is at least NM. The conclusion holds by substituting λ and K in (13). ■

Remark 2. To the best of our knowledge, the sample size formulas for SAA (Chapter 5, Shapiro et al. (2009)) are not intended for use to set up computational instances for solution. The reason for this is that the parameters necessary to apply measure concentration (e.g., Lipschitz constants, Diameter of the feasible set) are not available prior to machine preprocessing, and as a result, the bounds used in the calculations are poor, leading to overly conservative estimates of sample size. Instead, their primary role has been in showing that the growth of sample size for SAA depends logarithmically on the size of the feasible set and the reliability level $(1-\alpha)$. Our approach, seeking statistical optimality, allows us to estimate the reliability of a solution for during the course of a sampling-based algorithm and we report such computational results in section 6. ■

We make two other observations in connection with the vision of statistical optimality: (i) due to replications, there is a natural affinity towards parallel algorithms, and (ii) it promotes the use of adaptive solution algorithms which can increase the sample size of any replication without having to restart the algorithmic process from scratch. These properties are already available for SLP models through the SD algorithm.

5. Model Validation, Assessment and Selection

The stochastic programming (SP) literature has some foundational results for assessing solution quality as proposed in Mak et al. (1999). Shapiro and Homem-de Mello (1998) and Higle and Sen (1996a). However, these tests are not proposed within the larger context of model validation and assessment. The field of statistics, and more recently, Statistical Learning have developed notions of model selection on the basis of estimated errors for models which use empirical distributions. Because the LEO setup allows alternative plausible SL models, including the deterministic prediction model $\hat{m}$ as a potential alternative, it is important to adopt the SL approach of model validation, assessment and selection. Note that any model validation schema should depend on measuring the specific response function we wish to optimize. In this paper, our optimization objective reflects a sample average criterion. In cases where the optimization objective is not the expectation (such as robust optimization, mean-risk optimization), the validation methods proposed in this paper may not be appropriate.

The protocol we adopt is one based on Figure 1b where validation is critical part of the modeling process. In subsection 5.1 we discuss metrics for any LEO model, and comparisons between alternative LEO models will be presented in subsection 5.2. These tests correspond to the hypothesis tests used in the lower diamond-shaped block of Figure 1b, and require a decision as an input.

5.1. Model Validation and Prediction Interval

Identifying outliers is an integral part of regression modeling (James et al. (2013)), and a similar undertaking is advisable for the optimization setting. In this subsection, we will also identify prediction intervals to quantify the range of cost-to-go values which might result as a consequence of the decision.

Identifying Outliers. In stating the LEO model, the class of regressions $\mathcal{M}$ can be quite general. However, a model with Shared Spaces may call for a constrained regression where $\mathcal{M}$ may include bounds on predictions. For instance, in the LEO-Wyndor example, an unconstrained regression may lead to predictions which violate the bounds of the data-set. Unlike robust optimization where outliers may be critical to a decision model, our setting is more in line with regression model of statistics where the outliers can have a detrimental impact on the estimated conditional expectation. As in regression, where the focus is approximating a sample average prediction (using empirical residual minimization), data that are considered to be outliers should be removed. Similar considerations also hold for clustering algorithms (see Bertsimas and Shioda (2007)).

To identify outliers, it is important to choose the type of errors (additive scalar or multi-dimensional) to be used during cross-validation. Outliers from additive errors can be identified via Q-Q plots, whereas the case of multi-dimensional errors require greater computational effort. In cases such as MLR, the regression coefficients are allowed to be random variables, and hence lead to multi-dimensional error terms.

Outliers for an additive scalar model. Let $W_L = \min_i\{W_i : i \in T\}$ and $W_U = \max_i\{W_i : i \in T\}$. Once these bounds W_L, W_U have been computed, we identify those $i \in V$ as outliers by checking whether $m(Z_i, \xi) \in [W_L, W_U]$. Hence, data points with predictors outside the bounds ($[W_L, W_U]$) are considered to be ourliers because the response is far from the value predicted by the model. If outliers occurred due to incorrect recording during data collection, then we should remove them before SL modeling (see James et al. (2013)). Figure 3 shows the q-q plots for the error terms of the Training and Validation data sets of the LEO-Wyndor example before and after data preprocessing. We also compared the χ^2 test result of error sets before and after preprocessing. The detailed results of χ^2 test are included in section 6, where all computational results are presented.

An alternative way to identify outliers for the additive scalar SL model is to identify a $1-\alpha$ percent range where the parameter β_0 should belong. Choosing $\alpha = 0.05$, the acceptable range of β_0 is

$$\mathcal{E}_{M_0} = \Big\{ \beta_0 \;\Big|\; \frac{(\beta_0 - \bar{\beta}_0)^2}{s^2_{\beta_0}} \leq \chi^2(\alpha) \Big\},$$

where $\bar{\beta}_0$ denotes the mean value of the constant coefficient from MLR, and s_{β_0} is the standard deviation of the coefficient. We declare a data point (W_i, Z_i) to be an outlier if the following set is empty.

$$\begin{cases} (\bar{\beta}_0 + \xi_{i0}) + \bar{\beta}^\top Z_i = W_i \\ \bar{\beta}_0 + \xi_{i0} \in \mathcal{E}_{M_0} \end{cases}$$

The above test for uni-dimensional outliers is generalized to the multi-dimensional case below.

Outliers for a multi-dimensional model. Here we consider the case of statistical models in which the parameters are considered to be random variables. Given our current emphasis on MLR, we study the case of parameters which have multivariate normal distributions. For such statistical models, the Mahalanobis distance is known to provide a justifiable test for identifying outliers (Kulis et al. (2013)). The essence of our test revolves around identifying parameters β_i, which are expected to belong to a multi-dimensional ellipsoid

$$\mathcal{E}_M = \{ \beta \mid (\beta - \bar{\beta})^\top \Sigma_\beta^{-1} (\beta - \bar{\beta}) \leq \chi^2_p(\alpha) \},$$

where $\bar{\beta}$ denotes the mean value of coefficients reported for MLR, Σ_β is the variance-covariance matrix associated with the coefficients, p denotes the degrees of freedom for the χ^2 distribution and $1-\alpha$ represents the level of probability. If for a given data point (W_i, Z_i), the following system is infeasible, then we declare such a data point as an outlier.

$$\begin{cases} (\bar{\beta}_0 + \xi_{i0}) + (\bar{\beta} + \xi_i)^\top Z_i = W_i \\ (\bar{\beta}_0 + \xi_{i0}, \bar{\beta} + \xi_i) \in \mathcal{E}_M \end{cases}$$

These multi-dimensional feasibility problems are best solved in parallel using a convex optimization solver for each i.

Prediction Intervals. Under uncertainty, decision-makers (DM) not only seek a recommendation for a well-hedged decision, but they are interested in predictions of future costs. The 95% prediction interval, which is a population property, is most commonly used when one is interested in making inference on a single future observation of total cost of the decision model. We recommend prediction intervals which represent a non-parametric interval estimate of the population value of the cost-to-go-function at a 95% level. Unlike confidence intervals (even prediction intervals in regression), non-parametric prediction intervals may not be symmetric. The prediction interval can be obtained using the shortest interval which covers 95% of the validated cost-to-go population (see Theorem 1 in Frey (2013)). As noted in the cited paper, the coverage probability of the shortest interval is accurate for a uniform distribution, whereas, it is a lower bound on coverage probability for other distributions.

While the problem of seeking the shortest interval can be stated as an extension of a knapsack problem, we propose a somewhat tighter formulation (with additional valid inequalities). Let $\{h_j\}_{j=1}^{|V|}$ denote the validated cost-to-go data points, where $|V|$ represents the sample size of validation dataset. We recommend that the dataset be sorted increasing order (i.e., $h_j \leq h_{j+1}, \quad j = 1, ..., |V|-1$). Let z_j denote a binary variable which assumes a value 1 if h_j is included in the prediction interval, and 0 otherwise. In addition, define a binary variable u_j (v_j) which assumes a value 1 if h_j is the smallest (largest) index to be included in the prediction interval, and 0 otherwise. The problem identifying a $1-\alpha$ prediction interval is formulated as follows.

$$
\begin{aligned}
\min \; & \sum_{j=1}^{|V|} h_j v_j - \sum_{j=1}^{|V|} h_j u_j \\
\text{s.t.} \; & \sum_{j=1}^{|V|} v_j = 1, \quad \sum_{j=1}^{|V|} u_j = 1, \quad \sum_{j=1}^{V} z_j \geq (1-\alpha)|V| \\
& v_j \geq z_{j-1} - z_j, \quad u_j \geq z_j - z_{j-1}, \; j = 1, ..., |V| \\
& u_j \leq 1 - v_{j-t}, \quad t = 0, ..., j-1, \; j = 1, ..., |V| \\
& \sum_{t \leq j-1} u_t \geq v_j, \quad j = 2, ..., |V| \\
& z_t \leq 1 - u_j, \quad z_\tau \leq 1 - v_j, \; t = 0, ..., j-1, \; \tau = j+1, ..., |V|, \; j = 1, ..., |V|, \\
& z_j, u_j, v_j \in \{0, 1\}, \quad j = 1, ..., |V|.
\end{aligned}
$$

The $1-\alpha$ prediction interval is $\left[\sum_{j=1}^{|V|} h_j u_j, \sum_{j=1}^{|V|} h_j v_j\right]$. In the interest of brevity, we leave the interpretation of the formulation to the reader.

5.2. Comparison across LEO Models

In this subsection, we discuss how alternative LEO models are assessed and which of these should be recommended as the most appropriate. In order to do so, we estimate generalization error and optimization error.

Generalization Error. This quantity is a prediction of out-of-sample cost-to-go error in prescriptive models which may be observed when the system is implemented for a given decision x. Note that in our approach the generalization error is estimated based on the cost-to-go function values of SP models. The term "in-sample" includes all training as well as the validation data. For a given decision x, define

$$\begin{aligned} h_i^+ &\equiv \text{new value of the cost-to-go function} \\ \hat{h}_i &\equiv \text{a cost-to-go function value in the } \underline{\text{training dataset}} \text{ of sample size } |T| \\ h_i^v &\equiv \text{a cost-to-go function value in the } \underline{\text{validation dataset}} \text{ of sample size } |V| \end{aligned}$$

The random variable corresponding to h_i^+ (h_i^v) will be denoted as h^+ (h^v). For a given decision x, the new observation of the cost-to-go function can be estimated by the k nearest neighbors (k-NN) approximation from the training set. For a new observation (Z^+, W^+), define $\Delta_j^+ = ||Z_j - Z^+||$ where $j \in T$ and let $\Delta_{[j]}^+$ denote the j^{th} order-statistic of $\{\Delta_j^+, j \in T\}$. Let the indicator variable $I_{j,+}^k$ be 0 if $\Delta_j^+ > \Delta_{[k]}^+$, and 1 otherwise. Then the k-NN estimate for the case of disjoint spaces can be approximated as

$$h^+ = \frac{1}{k} \sum_{j=1}^{N} I_{j,+}^k h(x, \xi_j | Z = Z_j), \tag{14}$$

where $N = |T|$. For the case of shared spaces, we put the new data would take the form (x, W^+), hence we would use $Z^+ = x$ to compute the nearest neighbors. In keeping with the Lemma 1 of Walk (2008), we make the following assumption.

Assumption 4 (A4). Assume that $k/N \to 0$.

Define an approximation of the in-sample cost-to-go error as

$$\text{Err}_{in} \approx \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_{h^+} (h_i^+ - \hat{h}_i)^2. \tag{15}$$

The approximate equality ($\approx$) in (15) is intended to convey that the right hand side is asymptotically equal to the left hand side as N approaches infinity. In any event, the in-sample cost-to-go error provides an estimated average error between a new cost-to-go response and the training set cost-to-go.

Next we consider how well the training data predicts the cost-to-go values of the validation set. Similarly we provide a k-NN approximation of the cost-to-go values in the validation dataset for a given decision x. Let (Z^v, W^v) denote a validation data point, and let $\Delta^v_{[j]}$ denote the j^{th} order statistic of $\Delta^v_j, j \in T$ where $\Delta^v_j = ||Z_j - Z^v||$. Hence the k-NN approximation of the validation cost-to-go function value is given by

$$h^v = \frac{1}{k}\sum_{j=1}^{N} I^k_{j,v} h(x, \xi_j | Z = Z_j), \tag{16}$$

where the indicator variable $I^k_{j,v}$ is 0 if $\Delta^v_j > \Delta^v_{[k]}$, and 1 otherwise. Similar to the k-NN approximation in (14), we require A4 to be satisfied. Now define the cost-to-go training error (err_{tr}) as

$$\text{err}_{tr} = \frac{1}{N}\sum_{i=1}^{N}(h^v_i - \hat{h}_i)^2. \tag{17}$$

Given (15) and (17), the generalization error for the cost-to-go is estimated by $\mathbb{E}_h(\text{Err}_{in} - \text{err}_{tr})$. Then the following theorem suggests a mechanism to estimate this error. Although a similar result is available for the SL context of "goodness of fit" (Hastie et al. (2011)), it is not applicable because there is no parametric function to approximate the cost-to-go value. Since the setup for the "cost-to-go" function is based on a non-parametric estimate, we provide following result.

Theorem 3. Let A1 A2 and A4 hold. Then the generalization error for the cost-to-go is estimated by

$$\mathbb{E}_h(\text{Err}_{in}) - \mathbb{E}_h(\text{err}_{tr}) \approx \frac{2}{N}\sum_{i=1}^{N}\text{Cov}(h^v_i, \hat{h}_i). \tag{18}$$

Proof: Based on assumptions A1, A2 and A4, $\mathbb{E}_{h^+}(h^+_i)^2 = \mathbb{E}_h(h^v_i)^2$ and $\mathbb{E}_{h^+}h^+_i = \mathbb{E}_h h^v_i$. By the definitions of $\mathbb{E}_h(\text{Err}_{in})$ and $\mathbb{E}_h(\text{err}_{tr})$, the following equations hold:

$$\begin{aligned}\mathbb{E}_h(\text{Err}_{in}) - \mathbb{E}_h(\text{err}) &\approx \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_h\mathbb{E}_{h^+}(h^+_i - \hat{h}_i)^2 - \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_h(h^v_i - \hat{h}_i)^2 \\ &= \frac{1}{N}\sum_{i=1}^{N}\left[\mathbb{E}_h\mathbb{E}_{h^+}((h^+_i)^2 + \hat{h}^2_i - 2h^+_i\hat{h}_i) - \mathbb{E}_h((h^v_i)^2 + \hat{h}^2_i - 2h^v_i\hat{h}_i)\right]\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{N}\sum_{i=1}^{N}\left[\mathbb{E}_{h^+}(h_i^+)^2 - 2\mathbb{E}_{h^+}\mathbb{E}_h(h_i^+\hat{h}_i) - \mathbb{E}_h(h_i^v)^2 + 2\mathbb{E}_h(h_i^v\hat{h}_i)\right] \\
&= \frac{1}{N}\sum_{i=1}^{N}\left[2\mathbb{E}_h(h_i^v\hat{h}_i) - 2\mathbb{E}_{h^+}(h_i^+)\mathbb{E}_h(\hat{h}_i)\right] \\
&= \frac{1}{N}\sum_{i=1}^{N}\left[2\mathbb{E}_h(h_i^v\hat{h}_i) - 2\mathbb{E}_h(h_i^v)\mathbb{E}_h(\hat{h}_i)\right] = \frac{2}{N}\sum_{i=1}^{N}\mathrm{Cov}(h_i^v,\hat{h}_i).\blacksquare
\end{aligned}
$$

As is common in statistical learning, one obtains better estimation of generalization error by using cross-validation (Hastie et al. (2011)). In one run of cross-validation, the data is partitioned randomly into two complementary subsets. To analyze the generalization error for a given decision, we calculate the covariance of cost-to-go objectives from these independent subsets. Multiple runs of cross-validation will be performed to sample a generalization error, and finally, we report the estimate of the generalization error as the average value over k runs. Finally, note that if the decision model is a deterministic problem (the number of outcomes is 1), then the assumption A4 is not satisfied, and the estimate in (18) may only be a lower bound.

<u>Optimization Error.</u> To choose an optimum from all decisions, we need to find a proper metric to compare the estimated objectives among different models. In this case, we propose to undertake the Kruskal-Wallis test (Kruskal and Wallis (1952)), which does not assume normality as a condition for the test. The null hypothesis of the Kruskal-Wallis test is that the ranked medians of bins (of samples from two competing models) are the same. When the hypothesis is rejected, the cost-to-go values of one method stochastically dominates the cost-to-go of the other method.

Let $\tilde{f}_q$ denote the entire population vector of the validated objectives for model q, and let $\bar{f}_q$ denote the sample average of *validated objective* for model q. Then the model selection procedure is as follows. Let $\mathcal{Q} = \{1,\dots,Q\}$, choose small scalars $\alpha, \varepsilon_1, \varepsilon_2 > 0$, and do the following for $r \in \mathcal{Q}$:

$$
\text{if } \exists\ q \in \mathcal{Q} \ \text{ such that } \Big(\Pr\left(\|\tilde{f}_q - \tilde{f}_r\| > \varepsilon_1\right) \ge 1-\alpha \ \&\ \bar{f}_q \le \bar{f}_r - \varepsilon_2\Big), \mathcal{Q} \leftarrow \mathcal{Q}\setminus\{r\}. \tag{19}
$$

Denote $f^* = \min\ \{\bar{f}_q \mid q \in \mathcal{Q}\ \}$, we define the optimization error to be the difference $\bar{f}_q - f^*$ for any $q \in \{1,\dots,Q\}$. Note that the chance of committing a type I error increases when comparing many pairs of models. To prevent the inflation of type I error rates, it is also possible to use multiple cost-to-go data sets in a manner which only test the hypothesis that all cost-to-go data sets belong to the same distribution. However, such comparisons can only determine whether there is a difference in the median among all groups, but we cannot identify the winning model with such a procedure.

6. Illustrative Computations

We now turn to the LEO-ELECEQUIP and the LEO-Wyndor illustrations. All computations reported below are carried out by the SD algorithm because the models are SLPs.

6.1. LEO-ELECEQUIP

The specific model we solve is given in Appendix I. In this example, we use $c_u = 1, c_v = 3$ and $U_t = R_t = \infty$.

(a) Deterministic ARIMA Forcasting (DAF). Since U_t and R_t are infinity, we can use the predicted demand to define the order quantity as: $\Delta_t = \text{Max}\{0, \hat{D}_t - u_t\}$, where $\hat{D}_t$ is the expected value of the ARIMA model. This is a case of using $\hat{m}$ in section 3. We observe that current practice within many statistical consulting services adopt this process of using a deterministic forecast, once the time series model is built.

(b) Stochastic Linear Programming (SLP), which gives the decision by solving the instance in Appendix I (equation (20)). Note that our rolling horizon approach solves three period problems (0,1,2), and we use the solution of period 0 as our current decision, and drop the other decisions. We then use the demand of the following period, update the inventory status, and move the clock forward to the next period. This is a case of $m(z, \xi) = \hat{m}(z) + \xi$, where ξ is an outcome of $\tilde{\xi}$, the normal error from ARIMA.

6.1.1. Month by Month Validation Results for 2001-2002. The ARIMA model was trained on data from 1996-2000, and the performance of the models were validated during the two year period 2001-2002. Table 1 presents costs for the year 2001 and 2002 (24 months) for each of the two inventory policies DAF and SLP specified in (a) and (b) above. Note that of the 24 runs (simulating two years of inventory), the SLP approach cost higher only for month 1. Thereafter, it cost less in each subsequent month, with some (months) reducing costs by over 66%. The average inventory cost reduction over the deterministic ARIMA forecast is approximately 34% over the 2 year run. For practical decision-making the back-testing exercise performed above is more convincing, than any snapshot study of a dynamic process. Nevertheless, we present a snapshot of this inventory model in the interest of parsimony of presentation for both examples of this paper. However, we relegate these computational results to Appendix III.

Month	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
DAF Cost	12.33	14.41	39.02	14.54	26.44	28.86	7.65	37.99	25.26	38.46	16.92	30.34	11.35	3.05	15.11	26.74	15.67	38.98	33.23	23.81	17.90	16.62	15.31	29.72
SLP Cost	16.53	3.06	12.28	9.49	20.63	17.77	7.38	31.27	14.82	28.66	13.23	21.92	6.04	1.11	11.06	15.78	14.22	24.56	11.90	19.88	5.13	8.66	9.05	23.20

Table 1 LEO-ELECQUIP: Monthly Back-Testing Costs

6.2. LEO-Wyndor

We now present the LEO-Wyndor problem under alternative models. DF/LP represents learning enabled optimization using deterministic forecasts, in which we use the expected value of the linear regression as the demand model. This results in a deterministic LP. In addition, we also study other models where linear regression suggests alternative parameters: a) the additive scalar error model, using the empirical additive errors (EAE) and deterministic model coefficients $\beta_0, \beta_1, \beta_2$ where the first is the constant term, the second is the coefficient for TV expenditures, and the third is the coefficient for radio expenditures; b) a linear regression whose coefficients are random variables $\{\tilde{\beta}_j\}$, which are normally distributed and uncorrelated (NDU); c) a linear regression whose coefficients are random variables $\{\tilde{\beta}_j\}$ which are normally distributed and correlated (NDC). We reiterate that all three models EAE, NDU, NDC correspond to specific types of errors (which are indexed by q in the presentation in section 3). Note that for models NDU and NDC, we have continuous random variables, and as a result we adopted SD as the solution methodology because it manages discrete and continuous random variables with equal ease. We refer to these results by NDU/SD and NDC/SD. Also note that for the case of EAE, the dataset is finite and reasonably manageable. Hence we will use both SAA and SD for this model, and refer to them by EAE/SAA and EAE/SD.

6.2.1. Results for Error Terms. The calculations begin with the first test as the top diamond block in Figure 1b. Table 2 shows p-values and test results of χ^2 test for NDU/SD, NDC/SD and EAE. From values reported in Table 2, the fit appears to improve when a few of the data points near the boundary are eliminated. (see Figure 3).

	NDU/SD	NDC/SD	EAE
Before Data Preprocessing	0.44, not rejected	0.42, not rejected	0.45, not rejected
After Data Preprocessing	0.59, not rejected	0.57, not rejected	0.78, not rejected

Table 2 LEO-Wyndor: Comparison of Chi-square test

6.2.2. Results for Decisions and Optimal Value Estimates. The decisions and various metrics discussed earlier are shown in Table 3. Although the prediction interval is listed in a symmetric way, the actual data points are asymmetric with respect to the estimated mean. The last two rows report the probability γ and the corresponding tolerance level δ, which are provided by SD algorithm based on the theorems in section 4. We choose 1% of the mean value of validated objective to be δ_u in Theorem 2. Once again, notice that for both DF/LP and EAE/SAA, we do not report any probability because we use a deterministic solver as in (6).

The hypothesis test results for the cost-to-go objectives (the lowest diamond in Figure 1b) for each model are reported in Table 4. The T-test rejects the DF/LP model, where the others were not. The next two rows give the test results of variance based on F-statistic, and we conclude that none of the models can be rejected. We also performed a χ^2 test for the cost-to-go objectives using the training and validation sets. Again, the DF/LP model was rejected where as the others were not.

Remark 3. The concept of cross-validation (k-fold) is uncommon in the stochastic programming literature. With $k > 1$, this tool is a computational embodiment of (15), and provides a prediction of the error. Without such cross-validation, it is often likely that model assessment can go awry. For instance, in this example we have observed that if we use $k = 1$, then the EAE/SAA model can get rejected although using $k = 5$, the EAE/SAA is no longer rejected. This can be attributed to the fact that variance reduction due to $k = 5$-fold cross-validation reduces Type I error (when compared with $k = 1$). ■

Table 5 reports the optimization error, as well as the generalization error for all models. DF/LP shows the largest optimization error, which indicates that it is not an appropriate model to recommend for this application. On the other hand, NDU/SD and NDC/SD have comparable and relatively small generalization errors. However the optimization errors appear to be significant, therefore NDU/SD and NDC/SD do not yield the most profitable decisions. Had the criterion been simply the generalization error (as in SL), we might have chosen non-profit-maximizing decisions. In Table 6 we present the pairwise comparison of Kruskal-Wallis test. For the tests of DF/LP with other methodologies, the p-values are all smaller than 0.01, which implies that there are significant differences between the median ranks of DF/LP and each of the other four approaches. The stepped curve in Figure 2 illustrates the ordering discovered by the Kruskal-Wallis test. Note that DF/LP shows significant difference from the other approaches. Moreover, the curves for NDU/SD and NDC/SD are relatively close, whereas EAE/SAA and EAE/SD are indistinguishable. These similarities were quantified in Table 6 by the fact that the p-values for these comparisons are greater than 0.01. Finally, EAE/SAA and EAE/SD give the largest objective value, which is also reported in Table 3. The Kruskal-Wallis test suggests that the difference of EAE/SAA and EAE/SD is not significant, therefore both EAE/SAA and EAE/SD provide the most profitable decision.

7. Future Directions and Conclusions

In this paper, we introduced a fusion of concepts from SL and SP which is made possible by the manner in which we allow SL models to be imported as a model of uncertainty for decisions using SP. The interplay between SL and SP models may be used in a variety of ways, although we have only explored a couple of possibilities in this paper. One can easily envision other ways to integrate these types of models. For example, one might treat SL and SP models as agents in a game; another possibility is to introduce algorithmic interactions between SL and SP so that the search directions from one optimization (SL or SP) can immediately impact search directions of the other.

Models	DF/LP	NDU/SD	NDC/SD	EAE/SAA	EAE/SD
x_1	173.48	181.70	181.40	191.27	191.40
x_2	26.52	18.30	18.60	8.73	8.60
Estimated Obj.	\$41,391	\$41,580	\$41,492	\$42,009	\$42,045
Validated Obj. 95% C.I.	\$39,869(±692)	\$41,903 (±335)	\$41,865 (±302)	\$42,269 (±522)	\$42,274 (±493)
Validated Obj. 95% P.I.	\$39,856(±1,302)	\$41,911 (±632)	\$41,841 (±639)	\$42,258 (±1,012)	\$42,279 (±973)
Probability (γ)		0.9633	0.9698		0.9872
Tolerance (δ=percentage(value))		0.760%(316)	0.694%(288)		0.842%(357)

Table 3 LEO-Wyndor: Comparison of Solutions for Alternative Models

Models	DF/LP	NDU/SD	NDC/SD	EAE/SAA	EAE/SD
T-statistics($t < 1.96$)	$t = 2.18$	$t = 0.72$	$t = 0.84$	$t = 0.62$	$t = 0.49$
Cost-to-go Test(Mean)	rejected	not rejected	not rejected	not rejected	not rejected
F-statistics($0.67 < f < 1.49$)	$f = 1.23$	$f = 1.43$	$f = 1.29$	$f = 0.79$	$f = 1.16$
Cost-to-go Test(Variance)	not rejected	not rejected	not rejected	not rejected	not rejected
χ^2 Test p-value ($p > 0.05$)	$p = 0.038$	$p = 0.34$	$p = 0.32$	$p = 0.42$	$p = 0.42$
Cost-to-go Test(Distribution)	rejected	not rejected	not rejected	not rejected	not rejected

Table 4 LEO-Wyndor: Hypothesis Test Results for Alternative Models

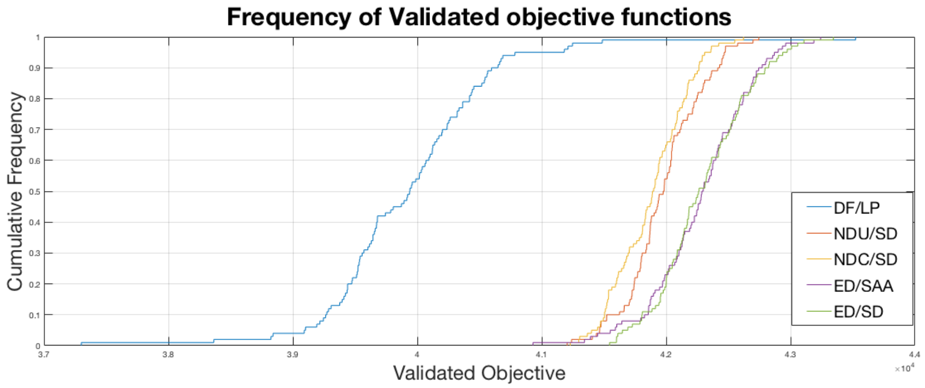

Figure 2 LEO-Wyndor: Stochastic Dominance of Validated Objectives under Alternative Models

Models	DF/LP	NDU/SD	NDC/SD	EAE/SAA	EAE/SD
Optimization Error	2405	371	409	5	
Generalization Error	29.751	19.406	19.554	21.889	21.326

Table 5 LEO-Wyndor: Errors for Alternative Models

Models	EAE/SD	EAE/SAA	NDC/SD	NDU/SD
DF/LP	2.76×10^{-8}	1.34×10^{-7}	1.12×10^{-7}	5.60×10^{-7}
NDU/SD	8.46×10^{-7}	6.2×10^{-3}	0.37	
NDC/SD	2.05×10^{-7}	1.72×10^{-3}		
EAE/SAA	5.87×10^{-2}			

Table 6 LEO-Wyndor: Kruskal-Wallis Test Results ($p > 0.01$)

To conclude the paper, let us revisit the four critical questions posed in the introduction.

1. (Modeling) In this paper, we have taken the first steps towards the fusion of SL and SP, thus enhancing the modeling capabilities for both. We discussed two classes of LEO models, and note that the one with disjoint spaces can allow very general SL models (e.g., ARIMA), while those with shared spaces call for more nuanced usage. Our illustrations also demonstrate the power of using stochastic functions for forecasting in SP over deterministic forecasts (resulting in LPs). Nevertheless, there are many open opportunities to explore.

2. (Statistical Optimality) We offered a new notion which seeks decisions with high probability of being within a given tolerance of the optimal value. The new optimality criterion also motivated using replications (in parallel, perhaps) to obtain probabilistic estimates of optimality. To the best of our knowledge, this is the first effort to report such statistically estimated metrics (probabilty of optimality) to support decision-making.

3. (Generalizability) Although we have borrowed the notion of generalizability from the SL world, the specific analysis for the cost-to-go estimate is in tune with SP. Our computational results demonstrate that although optimality is often the dominant criterion in an optimization model, one might be advised to choose one that has greater generalizability, when the optimality metrics appear to be similar.

4. (Model Validation and Selection) We introduced a model validation protocol for SP based on concepts of cross-validation in SL. In order to make this protocol realizable, we carried out statistical tests such as the Kruskal-Wallis test to identify which of the models provided the most desirable decision. In addition, we have also introduced a new metric for model assessment, namely the 95% Prediction Interval. Unlike the 95% C.I., this metric extends a similar metric used in regression analysis to give decision-makers a sense of the range of costs that may be incurred due to the selected decision. Such metrics are intended to give decision makers greater confidence in understanding the scope of costs that may be incurred due to uncertainty.

The novelty of these contributions is self-evident, not only from a conceptual (or theoretical) point of view, but also from modeling and computational perspectives. Using examples from the OR/MS discipline, we have shown how these ideas provide decision support which combines both statistical learning as well as stochastic programming. While our examples are drawn from linear models[4], the contributions summarized above have the potential to change the future of OR/MS modeling, especially for data-intensive and time-critical decision models. Such models are likely to include streaming and/or spatio-temporal data, and a variety of different classes of decision models (e.g., nonlinear, mixed-integer, dynamic and others). In particular, we plan to study extensions including a more direct coupling of SL and SP leading composite optimization. These will be particularly important to accommodate more general error distributions which do not satisfy homoscedasticity.

[4] combining multiple linear regression and stochastic linear programming

Appendix I: Details of Examples of Some LEO Models

The instances discussed below are developed using existing data-sets and existing optimization models. As with the rest of the paper, the novelty here is in the fusion of learning data-sets and optimization models. We include one example for each type of a LEO structure: Disjoint Spaces and Shared Spaces. Since the data-sets are not new, we append the acronym LEO to the names of the existing data-sets. Each model requires two aspects: one is the SL aspect, and the other is the decision/optimization aspect. In case of the SL part of the model, we assume that measures that are necessary to create acceptable SL models are undertaken. For the models given in this appendix, the time series setting (LEO-`ELECEQUIP` Inventory model) calls for stationarity tests, whereas in case of cross-sectional data (LEO-Wyndor model), outliers in the data should be identified and removed.

I.1. A Model with Disjoint Spaces: LEO-`ELECEQUIP` (Time-Series Data)

This model is based on a "rolling-horizon" decision model commonly used in inventory control. Before starting a sequence of decisions, one typically analyzes historical demand. In this particular example, we use a commonly available data set referred to as `ELECEQUIP` which provides demand of electrical equipment over a ten-year period. We will use the first five years to discover the time series pattern of demand, and then, use it within a rolling horizon inventory model. We conduct the validation exercise for two years, 2001-2002.

When fitting an ARIMA model to the training data, one identifies (p, d, q) and $(P, D, Q)_\tau$ where τ represents the seasonal backshift, and (p, d, q) specify the AR(p), I(d) and MA(q) components of ARIMA. In order to ascertain whether a series is stationary, it is customary to create an autocorrelation function (ACF) plot, and then one examines whether such a plot drops off to zero relatively fast. If so, we venture to guess that the data are stationary. While checking for correlation is not a substitute for independence, the lack of correlation can be considered as a surrogate.

<u>Model Details</u>: Without loss of generality, we can view the decision epoch as $j = 0$, and the most recent demand will be denoted $d_0 = D_j$ (from the validation data set). The experiment is over a two-year period, the decision epoch $j = 0, ..., 23$. In each decision epoch, we have the following "rolling-horizon" decision model.

The beginning inventory y_0 and and ending inventory u_0 are also available. The inventory model will look ahead into periods $t = 0, \dots, T,$, although as in rolling horizon schemes, only Δ_0 will be implemented. The model will be a two-stage multi-period stochastic program with the first stage making ordering decisions $x = (\Delta_0, \ldots, \Delta_{T-1})$, and the second stage predicting the potential cost of the choice Δ_0. As the decision clock moves forward in time, the total cost of inventory management becomes estimated by this process. The various relationships in the inventory model are summarized below.

- Because of the delivery capacity U_t, we must have $0 \leq \Delta_t \leq U_t$.
- We will sample demand realizations in period t using a forecast $D_t(\xi)$ (using the time series model created in the training phase). Here the notation ξ denotes one sample path of demands over the periods $t = 1, \ldots, T-1$. The notation d_0 (in lower case) denotes a deterministic quantity, whereas, the notation $D_t(\xi)$ denotes the outcome ξ of the demand (stochastic process) observed in period t. The planning horizon used in the linear program will be $T = 3$, which requires us to simulate two periods of demand.
- We assume that y_0 and d_0 are given. Let $u_t(\xi)$ denote the ending inventory in period t, and $y_{t+1}(\xi)$ denote the beginning inventory in period $t+1$. We have $y_{t+1}(\xi) = u_t(\xi) + \Delta_t$, and a storage (refrigerator) capacity constraint requires that $y_{t+1}(\xi) \leq R_{t+1}$, where the latter quantity is given. Then the ending inventory of period t, denoted $u_t(\xi)$, must obey the relationship $u_t(\xi) = \text{Max}\{0, y_t(\xi) - D_t(\xi)\}$. The unit cost of holding inventory is c_u, where $c_u \geq 0$. The total inventory holding cost for period t is then $c_u u_t(\xi)$.
- Let $v_t(\xi)$ denote the lost sales in period t, so that $v_t(\xi) = \text{Max}\{0, D_t(\xi) - y_t(\xi)\}$. Suppose that the per unit cost of lost sales in period t is c_v, where $c_v \geq 0$. Then the total cost of lost sales for period t is $c_v v_t(\xi)$, and the first stage cost is zero. Recalling that $x = (\Delta_0, \ldots, \Delta_{T-1})$, the cost-to-go function is defined as follows.

$$\text{Min}_{0 \leq \Delta_t \leq U_t} \; f(x) = \mathbb{E}_{\tilde{\xi}}\Big[h(x,\xi) = \sum_{t=0}^{T-1} c_u u_t(\tilde{\xi}) + c_v v_t(\tilde{\xi})\Big] \tag{20a}$$

$$\text{s.t.} \quad y_0 = d_0 \tag{20b}$$

$$y_{t+1}(\xi) - u_t(\xi) = \Delta_t \quad \text{for almost all } \xi \tag{20c}$$

$$y_{t+1}(\xi) \leq R_{t+1} \quad \text{for almost all } \xi \tag{20d}$$

$$-y_t(\xi) + u_t(\xi) \geq -D_t(\xi) \quad \text{for almost all } \xi \tag{20e}$$

$$y_t(\xi) + v_t(\xi) \geq D_t(\xi) \quad \text{for almost all } \xi \tag{20f}$$

$$u_t(\xi), v_t(\xi) \geq \mathbf{0} \tag{20g}$$

Note that constraints in (20) should be imposed for all possible errors in the training set. However, not all error combinations are sampled, and as result, we say that the constraints must hold for a large enough sample size (which is what we mean by the phrase "almost all" ξ). It suffices to say that the sample size used in optimization is decided during the Stochastic Decomposition (SD) algorithmic process. The computational results for T=2 are presented in the body of this paper.

I.2. A Model with Shared Spaces: LEO-Wyndor (Cross-Sectional Data for Production - Marketing Coordination)

We study a "textbook"-ish example which has been amalgamated from two textbooks: one on Operations Research (Hillier and Lieberman (2012)) and another on Statistical Learning (James et al. (2013)). Consider a well known pedagogical product-mix model under the banner of "The Wyndor Glass Co." In this example, Hillier and Lieberman (2012) address resource utilization questions arising in the production of high quality glass doors: some with aluminum frames (A), and others with wood frames (B). These doors are produced by using resources available in three plants, named 1, 2, and 3. The data associated with this problem is shown in Table 8.

	TV	Radio	Sales
1	230.1	37.8	22.1
2	44.5	39.3	10.4
3	17.2	45.9	9.3
$\cdots$	$\cdots$	$\cdots$	$\cdots$
200	232.1	8.6	13.4

Table 7 The Advertising Data Set (Source: James et al (2011)).

Plant	Prod. time for A (Hours/Batch)	Prod. time for B (Hours/Batch)	Total Hours
1	1	0	4
2	0	2	12
3	3	2	18
Profit	\$3,000	\$5,000	

Table 8 Data for the Wyndor Glass Problem (Hillier and Lieberman (2012)).

The product mix will not only be decided by production capacity, but also the potential of future sales. Sales information, however, is uncertain and depends on the marketing strategy to be adopted. Given 200 advertising time slots, the marketing strategy involves choosing a mix of advertising outlets through which to reach consumers. Exercising some "artistic license" here, we suggest that the advertising data set in James et al. (2013) reflects sales resulting from an advertising campaign undertaken by Wyndor Glass. That is, the company advertises both types of doors through one campaign which uses two different media, namely, TV and radio[5]. Note that in the original data set advertising strategy is represented as budgeted dollars, whereas we have revised it to represent advertising time slots. Thus in our statistical model, sales predictions are based on number of TV and radio advertising time slots[6]. In our interpretation, product-sales reflect total number of doors sold ($\{W_i\}$) when advertising time slots for TV is $Z_{i,1}$ and that for radio is $Z_{i,2}$, again as number of advertising time slots. (This data set has 200 data points, that is, $i = 1, \ldots, 200$). For the SP side, x_1 denotes the number of TV advertising time slots, and x_2 denotes the number of radio advertising time slots. The linear regression model for sales will be represented by $\hat{m}(x)$. We consider the following statistical models reported in Section 6.

Data Preprocessing

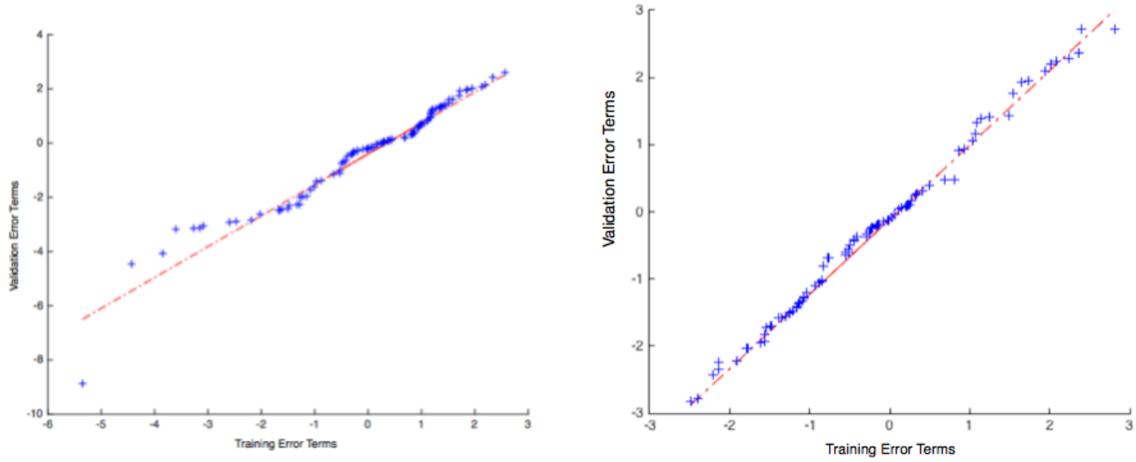

Figure 3 **q-q plot before and after data preprocessing**

1. (DF) For deterministic forecasts (DF) we simply use the sales given by $\hat{m}_1(z) = \bar{\beta}_0 + \bar{\beta}_1 z_1 + \bar{\beta}_2 z_2$. Thus, for deterministic predictions, we do not account for errors in forecast, whereas the remaining cases below use some form of error distributions.

2. (NDU) One approximation of the sales forecast is one where the correlation between the coefficients are ignored, and the resulting model takes the form $m_2(z, \xi) = (\bar{\beta}_0 + \xi_0) + (\bar{\beta}_1 + \xi_1)z_1 + (\bar{\beta}_2 + \xi_2)z_2$, where (ξ_0, ξ_1, ξ_2) are normally distributed and uncorrelated with mean zero, and the standard deviations are computed within MLR.

[5] The actual data set discussed in James et al. (2013) also includes newspapers. However we have dropped it here to keep the example very simple.

[6] The numbers used are the same as in James et al. (2013)

3. (NDC) Another approximation of the sales forecast is $m_3(z,\xi) = (\bar{\beta}_0 + \xi_0) + (\bar{\beta}_1 + \xi_1)z_1 + (\bar{\beta}_2 + \xi_2)z_2$, where (ξ_0, ξ_1, ξ_2) are <u>normally distributed and correlated</u> according to the variance-covariance matrix reported by MLR.

4. (EAE) This is the empirical additive error model, where $m_4(z,\xi) = \bar{\beta}_0 + \bar{\beta}_1 z_1 + \bar{\beta}_2 z_2 + \xi_0$, and ξ_0 denotes a random variable whose outcomes are $W_i - \hat{m}_4(Z_i)$. We refer to this model as <u>empirical additive errors</u>.

As in the set up for (6), the index q refers to the alternative error models (DF, NDU, NDC and EAE). The corresponding first stage is given as follows:

The formulation presented below mimics (6), and since all decisions variables x share the same space as the random variable Z, we explicitly remind the reader that $Z = z = x$.

<u>Index Sets and Variables</u>

$i \equiv$ index of product, $i \in \{A, B\}$.

$y_i \equiv$ number of batches of product i produced.

$$f(x) = -0.1x_1 - 0.5x_2 + \mathbb{E}[h(x,\ \tilde{\xi}_q \mid Z = z = x)] \tag{21a}$$

$$\text{s.t.} \quad x_1 + x_2 \leq 200 \tag{21b}$$

$$x_1 - 0.5x_2 \geq 0 \tag{21c}$$

$$L_1 \leq x_1 \leq U_1, L_2 \leq x_2 \leq U_2 \tag{21d}$$

$$h(x, \xi_q \mid Z = z = x) \quad = \quad \text{Max} \quad 3y_A + 5y_B \tag{22a}$$

$$\text{s.t.} \quad y_A \leq 4 \tag{22b}$$

$$2y_B \leq 12 \tag{22c}$$

$$3y_A + 2y_B \leq 18 \tag{22d}$$

$$y_A + y_B \leq m_q(z, \xi_q) \tag{22e}$$

$$y_A, y_B \geq 0 \tag{22f}$$

Note that the choice of ranges $[L_1, U_1]$ and $[L_2, U_2]$ are chosen so that assumption A2 is satisfied. Note that this instance is stated as a "maximization" model, whereas, our previous discussions were set in the context of "minimization". When interpreting the results, it helps to keep this distinction in mind. The LEO models presented above are relatively general, allowing very general regression models such as kernel-based methods, projection pursuit, and others. However, our current computational infrastructure is limited to stochastic linear programming (SLP) and as a result the regression used for models with Shared Spaces will be restricted to MLR.

Appendix II: Stochastic Programming Background - Sample Average Approximation (SAA) and Stochastic Decomposition (SD)

Sample Average Approximation(SAA)

Sample Average Approximation is a standard sampling-based SP methodology, which involves replacing the expectation in the objective function by a sample average function of a finite number of data points. Suppose we have sample size of K, an SAA example is as follows:

$$\min_{x \in X} F_K(x) = c^\top x + \frac{1}{K} \sum_{i=1}^{K} h(x, \xi^i). \tag{23}$$

As an overview, the SAA process may be summarized as follows.

1. Choose a sample size K, and sample K outcomes from the training data-set.

2. (Optimization Step). Create the approximation function $F_K(x)$, and solve an SAA instance (23).

3. (Validation Step). Take the decision from $F_K(x)$, follow the steps in section 5, estimate the validated confidence interval, generalization error and optimization error.

4. If the estimated objective does not agree with validated confidence interval, or generalization error and optimization error are not acceptable, increase the sample size K and repeat from step 1.

Assumption SAA-a. The expectation function $f(x)$ remains finite and well defined for all $x \in \mathbf{X}$. For $\delta > 0$ we denote by

$$S(\delta) := \{x \in \mathbf{X} : f(x) \leq f^* + \delta\} \quad \textit{and} \quad \hat{S}_K(\delta) := \{x \in \mathbf{X} : \hat{f}_K(x) \leq \hat{f}_K^* + \delta\},$$

where f^* denotes the true optimal objective, and f_K^* denotes the optimal objective to the SAA problem with sample size K.

Assumption SAA-b. There exists a function $\kappa : \Xi \to \mathbb{R}_+$ such that its moment-generating function $M_\kappa(t)$ is finite valued for all t in a neighborhood of zero and

$$|F(x', \xi) - F(x, \xi)| \leq \kappa(\xi) ||x' - x||$$

for a.e. $\xi \in \Xi$ *and all* $x', x \in \mathbf{X}$.

Assumption SAA-c. There exists constant $\lambda > 0$ *such that for any* $x', x \in \mathbf{X}$ *the moment-generating function* $M_{x',x}(t)$ *of random variable* $[F(x',\xi) - f(x')] - [F(x,\xi) - f(x)]$*, satisfies*

$$M_{x',x}(t) \leq exp(\lambda^2 ||x' - x||^2 t^2/2), \forall t \in \mathbb{R}.$$

From assumption SAA-b, $\Big|[F(x',\xi) - f(x')] - [F(x,\xi) - f(x)]\Big| \leq 2L||x' - x||$ *w.p. 1, and* $\lambda = 2L$.

Proposition 2. Suppose that assumptions SAA(a-c) hold, the feasible set $\mathbf{X}$ has a finite diameter D, and let $\delta_u > 0, \delta \in [0, \delta_u), \varepsilon \in (0,1)$, and $L = \mathbb{E}[\kappa(\xi)]$. Then for the sample size K satisfying

$$K(\varepsilon, \delta) \geq \frac{8\lambda^2 D^2}{(\delta_u - \delta)^2}\left[n \ln\left(\frac{8LD}{\delta_u - \delta}\right) + \ln\left(\frac{1}{1-\varepsilon}\right)\right], \tag{24}$$

we have

$$\Pr(\hat{S}_K(\delta) \subset S(\delta_u)) \geq \varepsilon.$$

Proof: This is Corollary 5.19 of Shapiro et al. (2009) with the assumption that the sample size K is larger than that required by large deviations theory (see 5.122 of Shapiro et al. (2009)). ■

Stochastic Decomposition (SD)

Unlike SAA which separates sampling from optimization, SD is based on sequential sampling and the method discovers sample sizes “on-the-fly” (Higle and Sen (1991),Higle and Sen (1994)). Because of sampling, any stochastic algorithm must contend with both variance reduction in objective values as well as solutions. SD uses M independent replications of value functions denoted f^ν, $\nu = 1, \ldots, M$. Each of these functions is a max-function whose affine pieces represent some Sample Average Subgradient Approximations (SASA). Because SD uses proximal point iterations, Higle and Sen (1994) shows that the maximum number of affine pieces is $n_1 + 3$, where n_1 is the number of first stage decisions, and these pieces are automatically identified using positive Lagrange multiplier estimates during the iterations. In theory, one can control the number of affine pieces to be smaller, but that can also be chosen ”on-the-fly” depending on the size of the first stage decision variables n_1. When the number of affine functions reduces to only 1, the SD method reduces to a proximal stochastic variance reduction method (prox-SVRG) (Xiao and Zhang (2014)).

Among other strengths such as parallelizability and variance reduction, one of the main usability issues which SD overcomes is that when a model has a large number of random variables (as in the case of multi-dimensional random variables) it does not require users to choose a sample size because the sequential process automatically discovers an appropriate sample. Together with the notion of Statistical Optimality as set forth in section 4, SD provides a very convenient optimization tool for LEO models.

For SLP models, Sen and Liu (2016) have already presented significant computational evidence of the advantage of SD over plain SAA. The reduced computational effort also facilitates replications for variance reduction (VR). VR in SD is achieved by creating the so-called compromise decision, denoted $\mathbf{x}^c$, which minimizes a grand-mean approximation $\bar{F}_M(x) := \frac{1}{M}\sum_{\nu=1}^{M} f^\nu(x)$, where $\{f^\nu\}_{\nu=1}^{M}$ denotes a terminal value function approximation for each replication m. Suppose that solutions $x^\nu(\varepsilon) \in (\varepsilon - \arg\min\ \{f^\nu(x) \mid x \in \mathbf{X}\})$ and $\mathbf{x}^c(\delta) \in (\delta - \arg\min\{\bar{F}_M(x) \mid x \in \mathbf{X}\})$. Then, Sen and Liu (2016) has proved consistency in the sense that $\lim_{\delta\to 0} \Pr(\bar{F}_M(\mathbf{x}^c(\delta)) - f^*) \to 0$. Here are the critical *assumptions* of SD (Higle and Sen (1996b)).

Assumption SD-a. The objective functions in the first and second stage models are either linear or quadratic, and all constraints are linear. Moreover, the set of first stage solutions is compact.

Assumption SD-b. The second stage optimization problem is feasible, and possesses a finite optimal value for all $x \in X$, and outcomes ξ (i.e., the relatively complete recourse assumption holds).

Assumption SD-c. The second stage constraint functions are deterministic (i.e., fixed recourse), although the right hand side can be governed by random variables. The set of outcomes of the random variables is compact.

Assumption SD-d. The recourse function h is non-negative. So long as a lower bound on the optimal value is known, we can relax this assumption. (Higle and Sen (1996b))

A high-level structure of SD algorithm can be summarized as follows. For greater specifics regarding two-stage SD, we refer to section 3.1 of Sen and Liu (2016), and for the multi-stage version we refer to Sen and Zhou (2014).

1. (Initialize). Let ν represent the number of completed replications, and set $\nu = 0$.

2. (Out-of-Sample loop). Set the number of completed replications $\nu = 0$. Increment ν at each time and start the next replication.

3. (In-Sample loop). Add one sampled outcome to the available samples and update the empirical frequencies.

4. (Updated Value Function Approximation). Using the new outcome from step 3 and previously generated approximations, update the new value function approximation $f_k^\nu(x)$.

5. (Optimization Step). Solve the regularization of $f_k^\nu(x)$ in step 4, and update an incumbent solution for the first stage.

6. (In-Sample Stopping Rule). If an In-Sample stopping rule is satisfied, output the incumbent solution $\mathbf{x}^\nu$ and continue to step 7. Else repeat from step 3.

7. (Out-of-Sample Stopping Rule). If the number of replications is greater than or equal to M, calculate a compromise decision $\mathbf{x}^c$ using a set of $\{\mathbf{x}^\nu\}_{\nu=1}^M$ and check whether $||\mathbf{x}^c - \bar{\mathbf{x}}||$ is within a pre-specified tolerance level. Else, repeat from step 2.

The value function approximation for replication ν is denoted f^ν and the terminal solution for that replication is x^ν. Note that we generate sample average subgradient approximations (SASA) using $K_\nu(\varepsilon)$ observations. Since these observations are i.i.d, the in-sample stopping rule ensures an unbiased estimate of the second stage objective is used for the objective function estimate at x^ν. Hence, the Central Limit Theorem (CLT) implies that $[K_\nu(\varepsilon)]^{\frac{1}{2}}[f(x^\nu) - f^\nu(x^\nu)]$ is asymptotically normal $\mathbf{N}(0, \sigma_\nu^2)$, where $\sigma_\nu^2 < \infty$ denotes the variance of $f^\nu(x^\nu)$. Since

$$N = \min_\nu K_\nu(\varepsilon), \tag{25}$$

it follows that the error $[f(x^\nu) - f^\nu(x^\nu)]$ is no greater than $O_p(N^{-\frac{1}{2}})$. The following result provides the basis for compromise solutions $\mathbf{x}^c$ as proved in Sen and Liu (2016).

Proposition 3. Suppose that assumptions SD(a-d) stated in the Appendix II hold. Suppose $\bar{\mathbf{x}}$ is defined as in Theorem 1, and $\mathbf{x}^c = \bar{\mathbf{x}}$. Then,

a) $\mathbf{x}^c$ solves

$$\operatorname*{Min}_{x \in X} \quad \bar{F}_M(x) := \frac{1}{M}\sum_{\nu=1}^{M} f^\nu(x), \tag{26}$$

b)

$$f(\mathbf{x}^c) \leq \bar{F}_M(\mathbf{x}^c) + O_p((NM)^{-\frac{1}{2}}), \tag{27}$$

c) $\mathbf{x}^c(\delta)$ denote an δ-optimal solution to (26). Let f^* denote the optimal value of the problem,

$$\lim_{\delta \to 0} ||\bar{\mathbf{x}}(\delta) - \mathbf{x}^c(\delta)|| \ \to \ 0 \text{ (wp1)}, \tag{28}$$

d)

$$\lim_{\delta \to 0} \Pr(|\bar{F}_{\delta,N}(\bar{\mathbf{x}}(\delta)) - f^*| \geq t) \ \to \ 0 \ \text{ for all } t \geq 0. \tag{29}$$

Proof: See Sen and Liu (2016).

Appendix III: Snapshot Computational Results for `ELECEQUIP`

A “snapshot” study for the inventory model provided in this Appendix. We select the end of the first year as the point in time when statistical comparisons are made. For this snapshot study, such a choice, allowing the model to run for a year, helps to avoid initialization bias.

Table 9 provides the estimated objective, validated objective (95% confidence and prediction intervals), and the solution quality. The probability of optimality reported in Table 9 is a result of the computations suggested in Theorem 2, where δ_u is chosen to be 1% of the total cost. Notice that we do not report a probability for the DAF model because it is simply a result of the ARIMA forecast. On the other hand, we include the probability for the SLP model, and this is consistent statistical optimality of section 4.

Table 10 summarizes results for three hypothesis tests for both DAF and SLP cases. A hypothesis test rejects the null hypothesis at the 95% level when the statistic lies outside the range provided in the table. Upon examining the entries for the T-test, the null hypothesizes for both DAF and SLP are not rejected. We also perform the F-test for DAF and SLP, and the F-test rejects the hypothesis that the variance of training and validation are the same at the 95% level. The results of the χ^2 test are presented in the last two rows, which analyzes the consistency of two data sets. Note that both DAF and SLP are not rejected at level $\alpha = 0.05$, but SLP shows a higher p-value. From these test results, we conclude that the SLP approach performs better in terms of consistency between training and validation sets.

Models	DAF	SLP
Estimated Obj.	25.52	22.75
95% Validated Confidence Interval	30.34(±3.84)	21.92(±3.38)
95% Validated Prediction Interval	30.87(±9.03)	21.75(±8.32)
Probability (γ)		0.9934
Tolerance (δ)		0.092

Table 9 **LEO-ELECQUIP: Comparison of Solutions under Alternative Models**

The comparison across models is provided in Table 11. The cost of SLP in validation is smaller than DAF by 9.42, and shows smaller generalization error as well. We include the $p-$value of Kruskal-Wallis test between DAF and SLP approaches, and the result shows that objectives of DAF and SLP methodologies have significantly different ranked medians. Since LEO-ELECEQUIP is a minimization problem, better solutions result in costs that are at the lower end of the horizontal (cost) axis. In this case, the better decision results from SLP, and Figure 4 gives evidence of this conclusion because for all cost levels C, the $Prob\ (Cost \leq C)$ is higher for SLP than it is for DAF.

Models	DAF	SLP
T-statistic ($t < 1.96$)	$t = 1.21$	$t = 0.20$
Cost-to-go Test (Mean)	not rejected	not rejected
F-statistic ($0.62 < f < 1.62$)	$f = 2.53$	$f = 1.23$
Cost-to-go Test (Variance)	rejected	not rejected
χ^2 Test p-value ($p > 0.05$)	$p = 0.13$	$p = 0.37$
Cost-to-go Test (Distribution)	not rejected	not rejected

Table 10 **LEO-ELECQUIP: Hypothesis Test Results under Alternative Models**

Models	DAF	SLP
Generalization Error	1.45	0.96
Kruskal-Wallis Test ($p-$value)	1.24×10^{-6}	
Optimization Error	9.42	

Table 11 **LEO-ELECQUIP: Errors under Alternative Models**

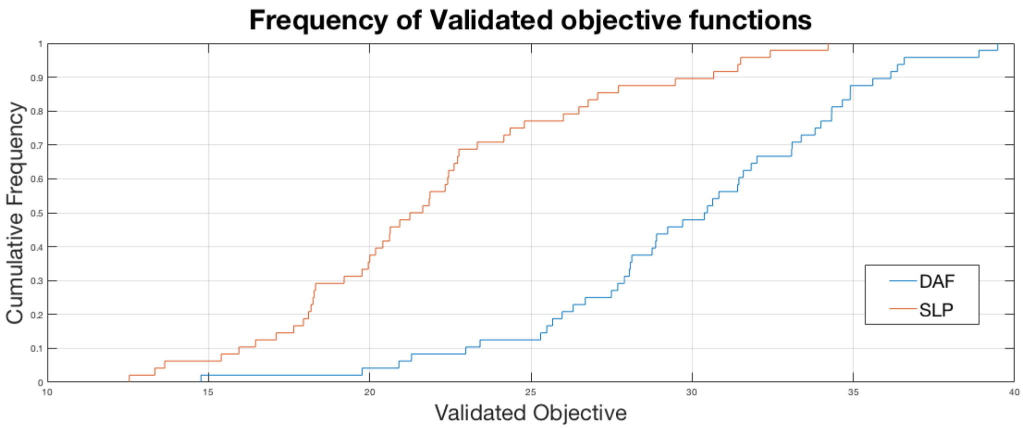

Figure 4 **LEO-ELECQUIP: Stochastic Dominance of SLP Validated objectives over DAF**

Acknowledgments

This research was supported by AFOSR Grant FA9550-15-1-0267, NSF Grant ECCS 1548847 and NSF Grant CMMI 1538605. We also thank Gabe Hackebeil for extending PySP functionality to allow SMPS output which is necessary for our SD code (available from the authors as well as the Github repository).

References

Ban, Gah-Yi, Noureddine El Karoui, Andrew E. B. Lim. 2017. Machine learning and portfolio optimization. *Management Science* doi:10.1287/mnsc.2016.2644.

Ban, Gah-Yi, Cynthia Rudin. 2017. The big data newsvendor: Practical insights from machine learning. *DSpace* .

Bayraksan, Güzin, David P Morton. 2011. A sequential sampling procedure for stochastic programming. *Operations Research* **59**(4) 898–913.

Bayraksan, Guzin, Péguy Pierre-Louis. 2012. Fixed-width sequential stopping rules for a class of stochastic programs. *SIAM Journal on Optimization* **22**(4) 1518–1548.

Ben-Tal, Aharon, Arkadi Nemirovski. 2001. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. SIAM.

Bertsekas, D.P. 2012. *Dynamic Programming and Optimal Control*. No. v. 2 in Athena Scientific optimization and computation series, Athena Scientific.

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

Bertsimas, Dimitris, Romy Shioda. 2007. Classification and regression via integer optimization. *Operations Research* **55**(2) 252–271.

Bertsimas, Dimitris, Melvyn Sim. 2004. The price of robustness. *Operations Research* **52**(1) 35–53.

Birge, John R, Francois Louveaux. 2011. *Introduction to Stochastic Programming*. Springer Science Business Media.

Blundell, Charles, Julien Cornebise, Koray Kavukcuoglu, Daan Wierstra. 2015. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424* .

Box, George EP, Gwilym M Jenkins, Gregory C Reinsel, Greta M Ljung. 2015. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons.

De Farias, Daniela Pucci, Benjamin Van Roy. 2004. On constraint sampling in the linear programming approach to approximate dynamic programming. *Mathematics of Operations Research* **29**(3) 462–478.

Frazier, Peter. 2012. Optimization via simulation with bayesian statistics and dynamic programming. *Proceedings of the Winter Simulation Conference*. Winter Simulation Conference, 7.

Freimer, Michael B, Jeffrey T Linderoth, Douglas J Thomas. 2012. The impact of sampling methods on bias and variance in stochastic linear programs. *Computational Optimization and Applications* **51**(1) 51–75.

Frey, Jesse. 2013. Data-driven nonparametric prediction intervals. *Journal of Statistical Planning and Inference* **143**(6) 1039–1048.

Friedman, Jerome H, Werner Stuetzle. 1981. Projection pursuit regression. *Journal of the American Statistical Association* **76**(376) 817–823.

Glynn, Peter W, Gerd Infanger. 2013. Simulation-based confidence bounds for two-stage stochastic programs. *Mathematical Programming* **138**(1-2) 15–42.

Hastie, Trevor J.., Robert John Tibshirani, Jerome H Friedman. 2011. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

Higle, Julia L, Suvrajeet Sen. 1991. Stochastic decomposition: An algorithm for two-stage linear programs with recourse. *Mathematics of Operations Research* **16**(3) 650–669.

Higle, Julia L, Suvrajeet Sen. 1994. Finite master programs in regularized stochastic decomposition. *Mathematical Programming* **67**(1-3) 143–168.

Higle, Julia L, Suvrajeet Sen. 1996a. Duality and statistical tests of optimality for two stage stochastic programs. *Mathematical Programming* **75**(2) 257–275.

Higle, Julia L, Suvrajeet Sen. 1996b. *Stochastic Decomposition*. Springer.

Hillier, Frederick S, G J Lieberman. 2012. *Introduction to operations research*. Tata McGraw-Hill Education.

Homem-de Mello, Tito, Güzin Bayraksan. 2014. Monte carlo sampling-based methods for stochastic optimization. *Surveys in Operations Research and Management Science* **19**(1) 56–85.

James, Gareth, Daniela Witten, Trevor Hastie, Robert Tibshirani. 2013. *An Introduction to Statistical Learning*, vol. 6. Springer.

Kao, Y-H., Benjamin V. Roy, Xiang Yan. 2009. Directed regression. Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, A. Culotta, eds., *Advances in Neural Information Processing Systems 22*. Curran Associates, Inc., 889–897. URL `http://papers.nips.cc/paper/3686-directed-regression.pdf`.

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.

Kruskal, William H, W Allen Wallis. 1952. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association* **47**(260) 583–621.

Kulis, Brian, et al. 2013. Metric learning: A survey. *Foundations and Trends® in Machine Learning* **5**(4) 287–364.

Linderoth, Jeff, Alexander Shapiro, Stephen Wright. 2006. The empirical behavior of sampling methods for stochastic programming. *Annals of Operations Research* **142**(1) 215–241.

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

Mak, Wai-Kei, David P Morton, R Kevin Wood. 1999. Monte carlo bounding techniques for determining solution quality in stochastic programs. *Operations Research Letters* **24**(1) 47–56.

Miller, Naomi, Andrzej Ruszczyński. 2011. Risk-averse two-stage stochastic linear programming: Modeling and decomposition. *Operations Research* **59**(1) 125–132.

Oliveira, Welington, Claudia Sagastizábal, Susana Scheimberg. 2011. Inexact bundle methods for two-stage stochastic programming. *SIAM Journal on Optimization* **21**(2) 517–544.

Pawlowski, Nick, Martin Rajchl, Ben Glocker. 2017. Implicit weight uncertainty in neural networks. *arXiv preprint arXiv:1711.01297* .

Powell, Warren B. 2011. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, vol. 842. John Wiley & Sons.

Rios, Ignacio, Roger JB Wets, David L Woodruff. 2015. Multi-period forecasting and scenario generation with limited data. *Computational Management Science* **12**(2) 267–295.

Royset, Johannes O, Roberto Szechtman. 2013. Optimal budget allocation for sample average approximation. *Operations Research* **61**(3) 762–776.

Royset, Johannes O, Roger JB Wets. 2014. From data to assessments and decisions: Epi-spline technology. *Tutorials in Operations Research: Bridging Data and Decisions*. INFORMS, 27–53.

Ruszczyński, Andrzej. 1986. A regularized decomposition method for minimizing a sum of polyhedral functions. *Mathematical programming* **35**(3) 309–333.

Ryzhov, Ilya O, Warren B Powell, Peter I Frazier. 2012. The knowledge gradient algorithm for a general class of online learning problems. *Operations Research* **60**(1) 180–195.

Sen, Suvrajeet, Yifan Liu. 2016. Mitigating uncertainty via compromise decisions in two-stage stochastic linear programming: Variance reduction. *Operations Research* **64**(6) 1422–1437.

Sen, Suvrajeet, Zhihong Zhou. 2014. Multistage stochastic decomposition: a bridge between stochastic programming and approximate dynamic programming. *SIAM Journal on Optimization* **24**(1) 127–153.

Shapiro, Alexander, Darinka Dentcheva, Andrzej Ruszczynski. 2009. *Lectures on Stochastic Programming*, vol. 10. SIAM, Philadelphia.

Shapiro, Alexander, Tito Homem-de Mello. 1998. A simulation-based approach to two-stage stochastic programming with recourse. *Mathematical Programming* **81**(3) 301–325.

Van Parys, Bart PG, Peyman Mohajerin Esfahani, Daniel Kuhn. 2017. From data to decisions: Distributionally robust optimization is optimal. *arXiv preprint arXiv:1704.04118* .

Walk, Harro. 2008. A universal strong law of large numbers for conditional expectations via nearest neighbors. *Journal of Multivariate Analysis* **99**(6) 1035–1050.

Xiao, Lin, Tong Zhang. 2014. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization* **24**(4) 2057–2075.