


Article

A Predictive Prescription Using Minimum Volume k -Nearest Neighbor Enclosing Ellipsoid and Robust Optimization

Shunichi Ohmori 

School of Creative Science and Engineering, Waseda University, Tokyo 169-8050, Japan; ohmori0406@aoni.waseda.jp; Tel.: +81-3-5286-3000

Abstract: This paper studies the integration of predictive and prescriptive analytics framework for deriving decision from data. Traditionally, in predictive analytics, the purpose is to derive prediction of unknown parameters from data using statistics and machine learning, and in prescriptive analytics, the purpose is to derive a decision from known parameters using optimization technology. These have been studied independently, but the effect of the prediction error in predictive analytics on the decision-making in prescriptive analytics has not been clarified. We propose a modeling framework that integrates machine learning and robust optimization. The proposed algorithm utilizes the k -nearest neighbor model to predict the distribution of uncertain parameters based on the observed auxiliary data. The enclosing minimum volume ellipsoid that contains k -nearest neighbors of is used to form the uncertainty set for the robust optimization formulation. We illustrate the data-driven decision-making framework and our novel robustness notion on a two-stage linear stochastic programming under uncertain parameters. The problem can be reduced to a convex programming, and thus can be solved to optimality very efficiently by the off-the-shelf solvers.

Keywords: robust optimization; predictive prescription; nonparametric model; convex optimization



Citation: Ohmori, S. A Predictive Prescription Using Minimum Volume k -Nearest Neighbor Enclosing Ellipsoid and Robust Optimization. *Mathematics* **2021**, *9*, 119. <https://doi.org/10.3390/math9020119>

Received: 27 November 2020
Accepted: 4 January 2021
Published: 7 January 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The term “analytics” was coined in a research report “Competing on Analytics” by Davenport (2006) [1] and has become widespread ever since. INFORMS defines analytics as “the scientific process of transforming data into insights for the purpose of making better decisions” [2]. With rapid progress in data-gathering technologies such as IoT (Internet of Things) and computation power, there are increasing expectations for the business analytics that will lead to the sophistication and automation of decision making in a rapidly changing and uncertain environment.

Business analytics is commonly viewed from three major perspectives: descriptive, predictive, and prescriptive (Lastig (2010) [3]; Evans (2012) [4]). The primal intent of descriptive analytics is to answer *what happened?* This includes preparing and analyzing historical data and identifying patterns from samples for reporting of trends. The primal intent of predictive analytics is to answer *what could happen?* This includes deriving *prediction of unknown parameters from data* using techniques such as statistics and supervised machine learning. The primal intent of prescriptive analytics is to answer *what should we do?* This includes deriving *decisions from known parameters* using techniques such as optimization.

These analytics techniques, however, are applied separately in most cases, and thus may end up in the suboptimal decision. Especially, it is not clear how the parameter prediction error in predictive analytics affects the decision-making in prescriptive analytics. For an optimization problem with uncertain parameters, it is reported that a parameter error of only 0.05% leads to a deterioration of the objective function value of 15–20% (Bental et al. 2009 [5]). Therefore, it is hard to say that it truly guides *decision from data*. Thus, how should we leverage data into decision-making?

Given these research gaps, in this research, we will clarify the methodology for deriving decisions from data by integrating the technologies of the predictive analytics and the prescriptive analytics. In order to achieve this research purpose, this research proposes an integrated framework of prediction algorithms and optimization algorithms as described below.

For the prediction algorithm, from the viewpoint of decision-making automation, it is desirable that the analyst can estimate from the data alone without assuming a prediction model formula. In this study, we applied the k -nearest neighbor method, which is one of the nonparametric regressions that can be estimated from the data without an explicit model assumption. The k -nearest neighbor method is one of the simplest algorithms, and is a method of averaging the closest k training data (k -nearest neighbors) in the feature space to perform prediction and discrimination. When the k -nearest neighbor is used for prediction, the average value of the data in the k -nearest neighbor is usually used. However, in the integrated framework proposed in this study, it is desirable to use all the samples in the k -nearest neighbor instead of a single prediction value in order to consider the robustness against the prediction error. Therefore, we propose a method to use a *predicted value set* without taking average of the samples in the k -nearest neighbors, as the input value of the optimization model.

For the optimization algorithm, robust optimization is applied in order to consider the prediction error of the data and to make it possible to calculate even large-scale data in a realistic time. The application of robust optimization requires the definition of an *uncertainty set* that indicates the possible range of data. This study proposes a method for finding the minimum volume ellipsoidal set including the predicted value set obtained by the k -nearest neighbors. As this problem is a convex planning problem, it is not affected by the so-called *curse of dimensionality* and can be solved very efficiently.

We call the proposed algorithm “a predictive prescription using minimum volume k -nearest-neighbor enclosing ellipsoid and robust optimization”. The novelty of this proposal is that it integrates predictive analytics and prescription analytics, considers prediction errors that could not be considered in the existing studies, and derives *decision from data*. Another novelty is to develop an algorithm that has few assumptions by analysts, is versatile, and has scalability that can withstand large-scale data. With this proposed technique, it is possible to achieve sophistication and automation of decision making using large-scale data, which is of great practical importance.

The remainder of the paper is as follows. In Section 2, we review the related research. In Section 3, we outline the modeling framework of the predictive prescription. In Section 4, we describe our proposed algorithm for predictive prescription using enclosing minimum volume ellipsoid for k -nearest neighbors. In Section 5, we illustrate the effectiveness of the proposed method over other predictive prescription approaches, with several numerical examples. In Section 6, we discuss potential extensions and variations.

2. Literature Review

In this section, we review related research. In Section 2.1, we review the stochastic programming and robust optimization, both of which are the framework for the decision-making under uncertainty. In Section 2.2, we review the research on the integration of machine learning and optimization. In Section 2.3, we state the contribution over the cited-research.

2.1. Stochastic Programming and Robust Optimization

In the field of prescriptive analytics, stochastic programming is widely studied as decision-making under uncertainty in parameters. In the stochastic programming, given the probability distribution of the unknown parameter, we seek the decision that minimizes the expected cost. In the real world, the probability distribution is unknown and must be inferred from the data. However, there are several difficulties as described below. First, when optimizing an unknown true objective function estimated from the observed data

that are subject to random error, even if the value estimates are unbiased, the uncertainty in these estimates coupled with the optimization-based selection process leads the value estimates for the recommended action to be biased high, which leads to that the resultant out-of-performance is often disappointing. This is called the *optimizer's curse* in the field of decision-making (Smith and Winkler 2006 [6]). Second, even if the probability distribution $p(u)$ and the decision x are given, the calculation of the expected value $\mathbb{E}(x, u)$ requires multiple integrals, which is # P-hard. Third, it is time-consuming to estimate the probability distribution, because it is necessary to make assumptions and validations about the statistical model several times. Given a need of quicker decision-making and shortage of data scientists, an autonomous data-driven decision-making has been of great practical interest. From this point of view, Birtsimas et al. (2019) [7] state that the probability distribution is an imaginary one derived from human assumptions and does not exist in reality. Only data exists in reality, and establishing a data-driven decision-making framework that does not explicitly assume a probability distribution is very important in today's data-rich world.

Robust optimization has been extensively studied as an alternative approach for the optimization under uncertainty. The key idea of the robust optimization is to define an uncertainty set as the possible range of the uncertainty parameter and minimize the worst-case objective function within the set (Bertsimas 2018 [8]). Charnes and Cooper [9] first proposed the chance constraints. Soyster (1973) [10] proposed the concept of uncertainty sets and solved the worst-case problems. Ben-Tal and Nemirovski [11–13], El-Ghaoui et al. (1997) [14], and El-Ghaoui et al. (1998) [15] derived a robust counterpart for a linear programming problem with ellipsoidal uncertainty and constructed a theory of robust optimization. Bertsimas and Sim (2004) [16] proposed the concept of *price of robustness* and considered ways to control conservatism. There are extensive review papers, see Ben-Tal et al. (2008) [17], Ben-Tal and Nemirovski (2009) [5], Gorissen et al. (2015) [18], Gabrel, Murat and Thiele (2014) [19], Sozuer and Thiele (2016) [20], and Delage and Iancu (2015) [21] and the references therein. However, in these studies, the optimization is performed under a given uncertainty set, therefore no data-driven mechanism is constructed.

In recent years, distributionally robust optimization (DRO) has also been widely studied, in which there is uncertainty in the probability distribution of parameters. Delge and Ye (2010) [22] proposed the DRO model with the moment-based ambiguity set. Ben-Tal et al. (2011) [23] proposed a robust discriminant analysis when there is uncertainty in the data. Dupacova and Kopa (2012) [24] studied the robustness of stochastic programming using the contamination method. Xu et al. (2012) [25] studied the probabilistic interpretation of robust optimization. They showed the connection between robust optimization and DRO, and showed that the solution of robust optimization is transformed to the solution of DRO. Zymler et al. (2013) [26] proposed an approximation of DRO using semi-definite programming. Wiesemann et al. (2014) [27] introduced an ambiguous set containing trust regions that can be represented in conic form. Ben-Tal et al. (2013) [28] proposed robust linear optimization problems with uncertainty regions defined by ϕ -divergences. Esfahani and Kuhn (2018) [29] proposed an ambiguity set derived from the Wasserstein distance. In these studies, the uncertainty region of the probability distribution of the parameter is inferred from the realization value of the parameter. However, the methodology for predicting parameters from data has not been clarified.

2.2. Integration of Machine Learning and Optimization

As mentioned above, predictive analytics has been studied in the field of statistics and machine learning (such as Melin and Castillo 2014 [30], Pozna and Precup 2014 [31], and Jammalamadaka et al. 2019 [32]), and prescription analytics has been studied in the field of mathematical optimization (e.g., Bertsimas et al. 2018 [8] and Esfahani and Kuhn 2018 [29]). In recent years, research on these integration has attracted attention.

Hertog and Postek (2016) [33] propose two opportunities to take advantage of the synergies of predictive analytics and predictive analytics. The first is the construction of a methodology for optimization using a predictive model. The second is the construction

of a methodology that automates optimization modeling by using predictive models. Elmachtoub and Grigas [34] have proposed a framework called Smart “Predict, then Optimize” (SPO). They proposed a prediction model that minimizes the SPO loss function, which minimizes the decision error, rather than the traditional prediction error. Larsen et al. (2018) [35] proposed a methodology for rapidly predicting solutions to discrete probabilistic optimization problems based on supervised learning. The training dataset consists of a number of deterministic problems that are solved independently and offline. Bertsimas et al. (2018) [36] and Dunn (2018) [37] proposed a tree-based algorithm called the optimal prescription tree (OPT). However, these approaches do not incorporate estimation errors into decision-making.

Recently, the study of the integration of predictive analytics and prescriptive analytics has been emerging. Among such, Bertsimas and Kallus (2019) [7] proposed the concept *predictive prescription*. This framework is very powerful, as it has two properties: asymptotic optimality and tractability. Asymptotic optimality ensures that as the number of samples approaches to infinity, the obtained solution approaches to the true optimal solution. Tractability ensures that the optimal solution can be computed in polynomial time and oracle calls, and, in many important cases, it is solvable using off-the-shelf optimization solvers.

For the extension of the model, Bertsimas (2017) [38] proposed the framework named “bootstrap robust analytics”, which integrates distributionally robust optimization and statistical bootstrap that are designed to produce out-of-samples guarantees by exploiting the use of a confidence region, derived from ϕ -divergence. Despite its fascinating property of bootstrap performance, the size of the inner maximization problem for the bootstrap robust formulation grows with the number of training data samples. Thus, finding a robust prescription may become computationally expensive when the training data set contains a huge amount of samples. They proposed dual formulation to ease the dependence on the amount of training data, which however is not completely eliminated.

2.3. Contribution

In this paper, we consider the research question centered around how to integrate predictive and prescriptive analytics. In order to fulfill the gap in the literature, several factors should be taken into account. The integration should be robust against the uncertainty of parameters caused by the prediction error. The integration should be distribution free. The integration should be computationally inexpensive when the training data set contains a huge amount of distinct samples. We propose an effective approach for a class of predictive prescription modeling that is tailored to uncertainty set constructions. The proposed algorithm utilizes enclosing minimum volume ellipsoid, which contains k -nearest neighbors of the observed auxiliary data. The proposed algorithm utilizes a nonparametric prediction model and thus does not need to assume probability distribution. The proposed algorithm forms around k -nearest neighboring samples and thus has robustness against the prediction error. The proposed algorithm utilizes robust optimization over ellipsoidal uncertainty, for which the efficient algorithm has been extensively studied. For linear programming (LP) under uncertain parameters, the problem can be reduced to a standard second-order cone programming (SOCP), and thus can be solved to optimality very efficiently by the custom solver. Therefore, the algorithm is computationally tractable.

The main contribution of the paper is as follows.

1. Most of the studies on the integration of machine learning and optimization use a separated approach, i.e., they predict uncertain parameters from auxiliary data first, then optimize with predicted uncertain parameters. This approach neglects the effect of prediction error, which is of critical importance in operations research and management science. We propose a framework that integrates machine learning and robust optimization to safeguard against the case when the estimation error yields serious trouble.

2. We make the nearest neighbor formulation advanced by Bertsimas and Kallus (2019) [7] resilient against the adverse effects of overfitting by formulating a robust counterpart. To form the robust counterpart, we propose the algorithm to form the minimum volume ellipsoid covering the k -nearest point. This ellipsoid is used as the *uncertainty set* in the robust counterpart. We indicate that the resulting robust supervised learning formulations are computationally as tractable as their nominal counterparts.
3. We demonstrate that the worst-case expectation over an ellipsoidal uncertainty set enclosing the k -nearest neighbor can in fact display good performance. We also investigate the out-of-sample performance of the resulting optimal decisions experimentally and analyze its dependence on the number of training samples and nearest neighbors.

3. Modeling Framework

This section describes the modeling framework. Table 1 presents a summary of the notation. In Section 3.1, the preliminary is explained. In Section 3.2, the predictive prescription is described. In Section 3.3, the alternative approach for the predictive prescription is described.

Table 1. Summary of notation.

Symbol	Description
n_u	dimension of uncertain parameter
n_v	dimension of auxiliary variable associate covariate
n_1	dimension of the 1st-stage decision
n_2	dimension of the 2nd-stage decision
M	the sample size of training data set
N	the sample size of validation data set
$u \in \mathbb{R}^{n_u}$	uncertain parameter
$v \in \mathbb{R}^{n_v}$	auxiliary variable associate covariate
$\mathcal{M}_T = \{(u_1, v_1), \dots, (u_M, v_M)\}$	training data set
$\mathcal{M}_V = \{(\hat{u}_1, \hat{v}_1), \dots, (\hat{u}_N, \hat{v}_N)\}$	test data set
$\mathcal{I}_T = \{j = 1, \dots, M\}$	index set for training data set
$\mathcal{I}_V = \{j = 1, \dots, N\}$	index set for test data set
$\bar{v} \in \mathbb{R}^{n_v}$	observation of v
$x \in \mathbb{R}^{n_1}$	1st-stage decision variable
$y \in \mathbb{R}^{n_2}$	2nd-stage decision variable
$f : \mathbb{R}^{n_1} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}$	1st-stage objective function
$g : \mathbb{R}^{n_2} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}$	2nd-stage objective function
$h : \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_v}$	predictor
c	cost vector of the 1st-stage decision variable
X	feasible region of x
$Y(u)$	feasible region of y given uncertain data u
\hat{z}_T	certificate (the objective function value of the data-driven solution, i.e., $f(\hat{x}_T)$)
\hat{x}_T	data-driven solution
\hat{z}_V	out-of-sample performance

3.1. Preliminary

In the predictive analytics, we seek predictor h to predict uncertain quantities of interest u (dependent variable) from associated covariates v (feature vector) as in (1)

$$u \sim \hat{u} = h(v), \tag{1}$$

given the *training dataset* $\{(u_1, v_1), \dots, (u_M, v_M)\}$, where u_j is the j -th observed data on u , and v_j is j -th observed data on v .

In the prescriptive analytics, we seek an optimal decision $x = [x_1, \dots, x_n]^T \in \mathbb{R}^n$ to be made, constrained in feasible region $x \in X$, so as to minimize some objective function $f(x, u)$ that depends on decision x and parameter u .

One possible way to incorporate the auxiliary data v on associate covariates into the model is to use supervised machine learning $\hat{u} = h(\bar{v})$ after observing $v = \bar{v}$, and solve the optimization problem as in (2).

$$\begin{aligned} & \text{minimize} && f(\hat{u}, x) \\ & \text{subject to} && x \in X \end{aligned} \tag{2}$$

This point-prediction approach, however, does not incorporate the uncertainty of the data, which is of critical importance in the business analytics. The traditional decision-making under uncertain data is stochastic-programming, which has the form of (3).

$$\begin{aligned} & \text{minimize} && \mathbb{E}_u f(u, x) \\ & \text{subject to} && x \in X \end{aligned} \tag{3}$$

If we know full joint distribution of u and v , say $p(u, v)$, we could incorporate uncertainty of u , utilizing the training dataset and observation $v = \bar{v}$ into the model. However, it is often difficult in practice to assume we have full knowledge of the joint distribution.

3.2. Predictive Prescription

In order to incorporate the auxiliary data v into the decision-making, we consider the predictive prescription model. The predictive prescription problem takes the form of (4).

$$\begin{aligned} & \text{minimize} && \mathbb{E}[f(x, u)|v = \bar{v}] \\ & \text{subject to} && x \in X \end{aligned} \tag{4}$$

In this framework, the objective is to minimize *conditional expected cost* wherein, on the basis of an observation of auxiliary covariates $v = \bar{v}$, a decision $x \in X$ is chosen in an optimal manner to minimize an uncertain cost $f(x, u)$ that depends on a random variable u .

In practice the joint distribution $p(u, v)$ is not known, and therefore must be inferred from data. This is called *data-driven settings*. In the data-driven settings, $p(u, v)$ is partially observable through a finite set of M independent samples, e.g., the training dataset $\mathcal{M}_T := \{(u_1, v_1), \dots, (u_M, v_M)\}$. In the training phase, the decision \hat{x}_T is obtained by minimizing the training problem (5).

$$\begin{aligned} & \text{minimize} && \mathbb{E}[f(x, u)|\mathcal{M}_T, v = \bar{v}] \\ & \text{subject to} && x \in X \end{aligned} \tag{5}$$

The solution of the training problem \hat{x}_T is called the *data-driven solution*, and the objective function value of the training problem $\hat{z}_T = \mathbb{E}[f(\hat{x}_T, u)|\mathcal{M}_T, v = \bar{v}]$ is called *certificate*.

The goal of a data-driven problem is to minimize *out-of-sample* performance of a data-driven solution \hat{x}_T is defined as in (6).

$$\hat{z}_V = \mathbb{E}_p f(\hat{x}_T, u) \tag{6}$$

As $p(u, v)$ is unknown, however, the exact out-of-sample performance cannot be evaluated in practice; therefore, it is evaluated by the *validation dataset* $\mathcal{M}_V = \{\hat{u}_1, \dots, \hat{u}_N\}$ as in (7).

$$\mathbb{E}_p f(\hat{x}_T, u) \simeq \frac{1}{N} \sum_{j=1}^N f(\hat{x}_T, \hat{u}_j) \tag{7}$$

We can extend the problem to a two-stage model, where the decision sequence is as follows. First, an observation of the auxiliary variable $v = \bar{v}$ is given. Second, the 1st stage *here-and-now* decision is made. Third, the realization of uncertain data u is given. Finally,

the 2nd stage *wait-and-see* decision is made. The two-stage predictive prescription model is formulated as in (8),

$$\begin{aligned} & \text{minimize}_x && c^T x + \mathbb{E}[f(x, u)|v = \bar{v}] \\ & \text{subject to} && x \in X \end{aligned} \tag{8}$$

where x is *here and now* variable, and $f(x, u)$ is the optimal value of the second-stage problem (9)

$$\begin{aligned} & \text{minimize}_y && g(y, u) \\ & \text{subject to} && y \in Y(u), \end{aligned} \tag{9}$$

and y is *wait and see* variable, g is objective function of second-stage problem, and $Y(u)$ feasible region of y given uncertain data u .

In the training phase, the input is training dataset \mathcal{M}_T and the output is decision \hat{x}_T and certificate $\hat{z}_T = c^T \hat{x}_T + \mathbb{E}[f(\hat{x}_T, u)|\mathcal{M}_T, v = \bar{v}]$. In the validation phase, the input is validation data \mathcal{M}_V , and the output is out-of-sample performance evaluated by $\hat{z}_V = c^T \hat{x}_T + \frac{1}{N} \sum_{j=1}^N f(\hat{x}_T, \hat{u}_j)$.

3.3. Alternative Approach

A natural approach to generate data-driven solutions \hat{x}_T is the sample approximate approximation (SAA) formulation that approximate p with p_N where $p_j = (1/M), \forall j \in \mathcal{M}_T$. SAA formulation with training samples u_j for the two-stage problem can be written as one large-scale problem (10)

$$\begin{aligned} & \text{minimize} && c^T x + \frac{1}{M} \sum_{j=1}^M p_j f(x, u_j) \\ & \text{subject to} && x \in X \end{aligned} \tag{10}$$

This can be written as an integrated form (11).

$$\begin{aligned} & \text{minimize} && c^T x + \frac{1}{M} \sum_{j=1}^M p_j g(y_j, u_j) \\ & \text{subject to} && x \in X \\ & && y_j \in Y(u_j), \quad j = 1, \dots, M \end{aligned} \tag{11}$$

This formulation, however, does not exploit auxiliary variable v . Another alternative approach is the point-prediction approach (12).

$$\begin{aligned} & \text{minimize} && c^T x + f(x, \hat{u}) \\ & \text{subject to} && x \in X, y \in Y(\hat{u}) \\ & && \hat{u} = h(\bar{v}) \end{aligned} \tag{12}$$

This approach can exploit the auxiliary variable. However, this approach does not consider the robustness against prediction error, as is stated in the single-stage problem. This may lead to the poor out-of-sample performance or the violation of feasibility with respect to $y \in Y(u)$.

Bertsimas and Kallus (2019) [7] proposed k -nearest neighbor formulation weight using kNN (13),

$$\pi_{knn}(\bar{v}) = \begin{cases} 1/k & \text{if } j \in \mathcal{N}_k(\bar{v}) \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

where $\mathcal{N}_k(\bar{v})$ is k -nearest points to the observed auxiliary variable $v = \bar{v}$. Using this weight, the two-stage predictive prescription can be transformed into the problem (14).

$$\begin{aligned} & \text{minimize} && c^T x + \sum_{j=1}^M \pi_{knn}(\bar{v}) f(x, u_j) \\ & \text{subject to} && x \in X \\ & && y \in Y(u_j), \quad j = 1, \dots, k \end{aligned} \tag{14}$$

We let $\hat{z}_{PR}, \hat{x}_{PR}$ be optimal value and solutions of the two-stage predictive prescription problem, respectively. \hat{x}_{PR} is proven to converge almost surely to their counterparts

of the true problem with $M \rightarrow \infty$. However, \hat{x}_{PR} tends to display a poor out-of-sample performance in situations where M is small and the acquisition of additional samples would be costly. Furthermore, the number of constraints grows with k , which is computationally challenging when k is large.

4. Proposed Algorithm

In this section, we describe the proposed algorithm. The proposed algorithm is outlined below. First, the k -nearest points are drawn from the training samples \mathcal{M}_T . Second, the minimum volume enclosing ellipsoid \mathcal{E}_{knn} that contains the finite set $\{u_j | \forall u_j \in \mathcal{N}_{knn}(\bar{v})\}$. Finally, the robust counterpart of the two-stage stochastic problem is solved. Each of these steps are described in detail in the following sections.

4.1. k -Nearest-Neighbor

The k -nearest-neighbor algorithm (kNN) is a nonparametric method used for classification and regression. k -nearest neighbor nonparametric regression method is a broadly applied algorithm, which has nonparametric, small error ratio, and good error distribution (Yiannis and Poulicos 1857 [39]). It is a nonparametric model, and thus the predictor does not take a predetermined form but is constructed according to information derived from the data.

$\mathcal{N}_k(\bar{v})$ contains the indices of the k closest points of v_1, \dots, v_k to \bar{v} . The distance can be measured in several ways. In this research, we use 2-norms, which is one of the most standard distance measures. A heuristically optimal number k of nearest neighbors can be found based on root mean square error (RMSE) using cross-validation.

In the standard kNN regression model, the uncertain parameter u can be predicted by taking average of the training samples in the k -nearest neighbors as in the Equation (15).

$$\begin{aligned} \hat{u} &= \pi_j^{knn} u_j \\ \pi_j^{knn} &= \begin{cases} 1/k & \text{if } j \in \mathcal{N}_k(\bar{v}) \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{15}$$

In the proposed algorithm, however, in order to have a safeguard against prediction error in the optimization model, the regression model is not used. Instead, we form the minimum volume ellipsoid enclosing all points in $\mathcal{N}_k(\bar{v})$, hoping that the uncertain parameter u lies in that ellipsoid.

4.2. Minimum Volume Ellipsoid Around a Set

In this section, we present an algorithm for computing the minimum-volume ellipsoid that must contain k -nearest neighbors to form the uncertainty set U .

We consider the problem of finding the minimum volume ellipsoid that contains the samples in k -nearest neighbor $\mathcal{N}_k(\bar{v})$. An ellipsoid covers $\mathcal{N}_k(\bar{v})$ if and only if it covers its convex hull, so finding the minimum volume ellipsoid that covers $\mathcal{N}_k(\bar{v})$ is the same as finding the minimum volume ellipsoid containing the polyhedron $\text{conv } \mathcal{N}_k(\bar{v})$. We parameterize the ellipsoid as in (16).

$$\mathcal{E}_{knn} = \{u | \|Pu + \rho\|_2 \leq 1\} \tag{16}$$

We can assume without loss of generality that P is positive semidefinite, in which case the volume of \mathcal{E}_{knn} is proportional to $\det P^{-1}$. The problem of computing the minimum volume ellipsoid containing $\mathcal{N}_k(\bar{v})$ can be expressed as in (17),

$$\begin{aligned} &\text{minimize} && \log \det P^{-1} \\ &\text{subject to} && \|Pu_j + \rho\|_2 \leq 1, \quad j = 1 \in \mathcal{N}_k(\bar{v}) \end{aligned} \tag{17}$$

where the variables are P and ρ , and the implicit constraint P is positive semidefinite. The objective and constraint functions are both convex in P and ρ , so the problem is convex. See Boyd (2004) [40] for further detail.

Once P and ρ are obtained, we transform from (P, ρ) to (R, \bar{u}) as in (18) and (19).

$$R = P^{-1} \tag{18}$$

$$\bar{u} = -R\rho \tag{19}$$

and we can form the uncertainty set derived from the k -nearest neighbor (20)

$$\mathcal{E}_{knn} = \{\bar{u} + R w \mid \|w\|_2 \leq 1\}, \tag{20}$$

where \bar{u} can be interpreted as nominal value of the uncertain parameter.

4.3. Robust Optimization

The robust optimization is the framework that random variables are modeled as uncertain parameter u belonging to a convex uncertainty set U and the decision-maker protects the system against the worst-case within that set. The robust optimization takes the form of (21).

$$\begin{aligned} & \text{minimize} && \sup_{u \in U} f(x, u) \\ & \text{subject to} && x \in X. \end{aligned} \tag{21}$$

The robust optimization can be transformed into the following inequality form (22).

$$\begin{aligned} & \text{minimize} && z \\ & \text{subject to} && \sup_{u \in U} f(x, u) \leq z \\ & && x \in X. \end{aligned} \tag{22}$$

In the proposed method, we use the minimum volume ellipsoid to cover the k -nearest neighbor to the observation $v = \bar{v}$, presented in the previous section, to form the uncertainty set. Ellipsoid uncertainty has been extensively studied for over a decade. One advantage of the ellipsoid uncertainty is that it is found to be not too pessimistic, compared to the box uncertainty $\|u\|_\infty \leq 1$. Another desirable feature is that the robust counterpart can be derived via the second-order cone programming, which can be solved very efficiently.

For a linear constraint in inequality form in which u is known to lie in given ellipsoids and the constraints must be satisfied for all possible values of the u

$$u^T x \leq 0, \quad \forall u \in \mathcal{E}_{knn} = \{\bar{u} + R w \mid \|w\|_2 \leq 1\}, \tag{23}$$

The *robust counterpart* of the inequality can be expressed as

$$\sup\{u^T x \mid u \in \mathcal{E}_{knn}\} \leq 0. \tag{24}$$

The left-hand-side can be expressed as

$$\begin{aligned} \sup\{u^T x \mid u \in \mathcal{E}_{knn}\} &= \bar{u}^T x + \sup\{u^T R^T x \mid \|u\| \leq 1\} \\ &= \bar{u}^T x + \|R^T x\|_2. \end{aligned} \tag{25}$$

Thus, the robust linear constraint can be expressed as second-order cone inequality.

4.4. Overall Algorithm

The overall Algorithm 1 is described as follows.

Algorithm 1 Summary of algorithm.

1. **pick** k -nearest points $\mathcal{N}_k(\bar{v})$
2. **form** minimum volume ellipsoid \mathcal{E}_{knn} to cover $\{j|\forall j \in \mathcal{N}_k(\bar{v})\}$

$$\mathcal{E}_{knn} = \{\bar{u} + R w \mid \|w\|_2 \leq 1\} \tag{26}$$

3. **solve** robust optimization problem

$$\begin{aligned} &\text{minimize} && c^T x + \sup_{u \in \mathcal{E}_{knn}} f(x, u) \\ &\text{subject to} && x \in X \\ &&& y \in Y(u), u \in \mathcal{E}_{knn} \end{aligned} \tag{27}$$

The proposed algorithm has desirable properties in the data-driven predictive prescription. First, the proposed algorithm utilizes nonparametric k -nearest neighbor and thus does not need to assume the joint probability distribution. Second, the proposed algorithm forms around k -nearest neighboring samples and thus has robustness against the prediction error. Third, the proposed algorithm utilizes robust optimization over ellipsoidal uncertainty, for which the efficient algorithm has been extensively studied, and therefore is computationally tractable.

5. Numerical Example

We demonstrate the effectiveness of the proposed method with numerical experiments. We restrict our discussion to a two-stage data-driven linear predictive prescription model. We applied and compared the following alternative approaches: the sample average approximation (SAA), the point-prediction (PP) approach, the predictive-prediction (PR) approach, and the proposed approach. First, we apply the proposed framework to a small-sized problem in which there are two variables and two constraints in the first and second stages and show how the proposed method is applied. Second, we expand the experiments with larger sized problems.

Experimental conditions are Intel(R) CoreTM i7-8700 (3.20 GHz, 3.19 GHz) with 32.0 GB memory. The program was coded in Julia with Gurobi optimizer called from Convex.jl.

5.1. Small-Size Instance

For the ease of exhibition, we consider a two-stage stochastic linear-programming problem (28)

$$\begin{aligned} &\text{minimize} && 3x_1 + 5x_2 + \mathbb{E}_u f(x, u) \\ &\text{subject to} && 2x_1 + x_2 \geq 3 \\ &&& x_1 + 3x_2 \geq 5 \\ &&& x_1 \geq 0, x_2 \geq 0 \end{aligned} \tag{28}$$

where $f(x, u)$ is the optimal value of the following 2nd stage problem, (29)

$$\begin{aligned} &\text{minimize} && 4y_1 + 6y_2 \\ &\text{subject to} && u_1 x_1 + u_2 x_2 + 2y_1 + 4y_2 \geq 6 \\ &&& y_1, y_2 \geq 0. \end{aligned} \tag{29}$$

Note that the problem has complete recourse, i.e., if there exists a solution y satisfying $u_1 x_1 + u_2 x_2 + 2y_1 + 4y_2 \geq 6$ and $y_1 \geq 0, y_2 \geq 0$ for every x .

We assume that u follows the multivariate normal distribution with known β and Σ as in (30)

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \mathcal{N}\left(\begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} v, \Sigma^2\right) \tag{30}$$

In this experiment, we set $[\beta_1, \beta_2]^T = [10, 30]^T$ and $\Sigma_{11} = 3000, \Sigma_{12} = \Sigma_{21} = 900, \Sigma_{22} = 600$. We also assume that v follows the normal distribution with known μ and σ as in (31).

$$v \sim \mathcal{N}(\mu, \sigma^2) \quad (31)$$

In this experiment, we set $\mu = 100$ and $\sigma = 30$.

We generate training samples \mathcal{M}_T and test samples \mathcal{M}_V . The decision-maker does not know the true distribution or test samples. The decision-maker only knows the training samples generated from the true distribution.

We have M training samples and N test samples. We repeat this same experiment where the decision-maker sees M samples and solves the problem 10,000 times. Each time, we record the optimal value of the optimization problem \hat{z} . Each time, we got an optimal decision \hat{x}_T which is a random variable that depends on training samples. Each of these decisions, we evaluated the objective of the optimization problem by using another N test samples to compute the out-of-sample performance. The optimal value will be random because the training samples are random.

Table 2 presents the average out-of-sample performance, where SAA, PP, PR, and RO denote the average out-of-sample performance of the sample average approximation, point-prediction approach, predictive prescription, and proposed method, respectively. From Table 2, we see that PP derived the worst out-of-sample performance of these. This is because the PP approach does not take into account the robustness against the prediction error. From the assumption that the joint distribution of u and v is a multivariate normal distribution, even if v is obtained, the range that u can take varies. The point prediction approach makes decisions without considering this variability, and as a result, the out-of-sample performance was disappointing. The SAA approach was the second worst, which was also disappointing. This is because SAA does not use the information of v , so regardless of the value of v , all samples are used as training data to make decisions. Therefore, it is considered that the overfitting made the out of sample performance worse because the possible ranges of training data and validation data are different. PR and RO, both of which utilized auxiliary data v and consider the robustness against prediction error, derived much better out-of-sample performance than the other two. Furthermore, the proposed method was able to obtain the best value of all. This is because the PR approach makes decisions using only the samples that appear in the k -nearest neighbor as training data, whereas in RO, the uncertainty set is defined using the minimum volume enclosing ellipsoid. Therefore, it is considered that the better result was obtained because the worst case is taken even for the unknown sample.

Table 3 presents the average out-of-sample performance of the proposed algorithm with the different number of training samples that included in the k NN. From the Table 3, it can be seen that the quality of the proposed method changes greatly depending on the value of k . As k increases, the robust optimization approach does not work well. To consider the reason for this, Figure 1 shows the minimum volume ellipsoid with $k = 10$ and $k = 50$, where the horizontal axis represents u_1 and the vertical axis represents u_2 , the blue dots indicate all training data, the black dots indicate validation data, the red dots indicate samples within the k -nearest neighbor, and the green line indicates the obtained minimum volume ellipsoid. From this Figure 1, we see that the distributions of training data and validation data are significantly different. At $k = 10$, it can be seen that the distribution of the samples in the k -nearest neighbor and the distribution of validation data are close. On the other hand, when $k = 50$, the distribution of validation data differs greatly from that of k -nearest neighbors because the k -nearest neighbors are too large. These results indicate that by setting k properly, the corresponding ellipsoid covers the proper size of uncertainty.

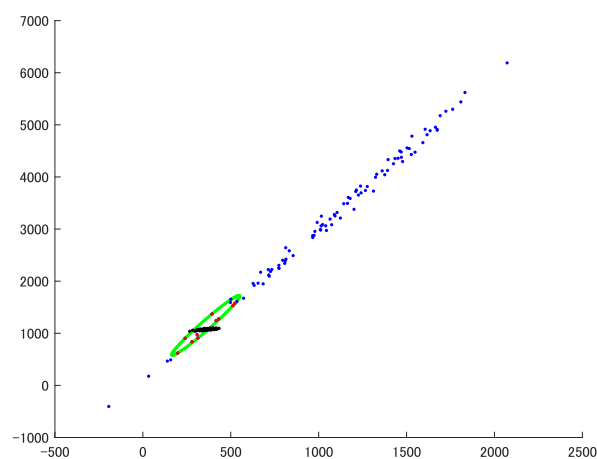
Table 2. Average out-of-sample performance ($k = 10$).

SAA	PP	PR	RO
40.2	111.3	23.9	12.5

Table 3. Average out-of-sample performance with different sample size in the k NN.

k	PP	PR	RO
10	111.3	33.4	12.5
20	173.3	40.9	37.1
30	230.6	32.4	30.6
40	240.5	28.1	35.0
50	327.5	26.0	119.2
60	396.3	25.1	134.2
70	446.7	24.5	209.7
80	475.6	23.9	263.8
90	566.1	37.0	497.3

(a) $k = 10$



(b) $k = 50$

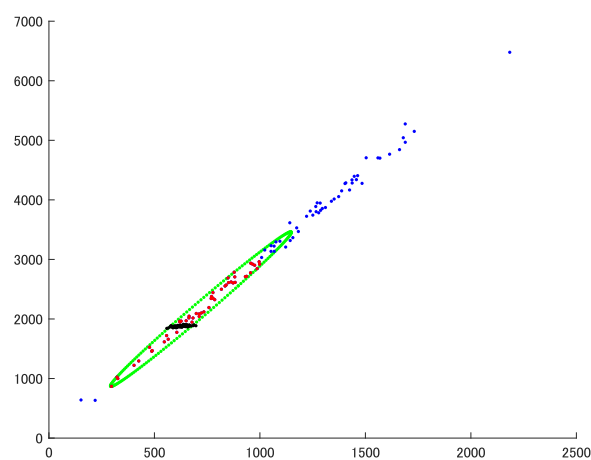


Figure 1. Minimum volume enclosing ellipsoid (horizontal axis: u_1 ; vertical axis: u_2).

5.2. Large-Size Instances

We consider a two-stage stochastic linear programming problem:

$$\begin{aligned} & \text{minimize} && c^T x + \mathbb{E}_u f(x, u | v = \bar{v}) \\ & \text{subject to} && Ax = b \end{aligned} \tag{32}$$

where $c \in \mathbb{R}^{n_1}$, $A \in \mathbb{R}^{n_1 \times m_1}$, and $b \in \mathbb{R}^{m_1}$ are first-stage parameters and $f(x, u)$ is an optimal value of the second-stage problem

$$\begin{aligned} & \text{minimize} && q^T y \\ & \text{subject to} && s_i x + t_i y \geq w_i, \quad i = 1, \dots, m_2 \end{aligned} \tag{33}$$

and $u := [q^T, s_1^T, \dots, s_{m_2}^T, t_1^T, \dots, t_{m_2}^T, w^T]^T$ with second-stage parameters $q \in \mathbb{R}^{n_2}$, $S \in \mathbb{R}^{m_2 \times n_1}$, $T \in \mathbb{R}^{m_2 \times n_2}$, $w \in \mathbb{R}^{m_2}$. The problem has complete recourse, i.e., there exists y that satisfying $Sx + Ty \geq w$ for $\forall x$.

We assume that $u := (q, S, T, w)$ follows the multivariate normal distribution as in (34).

$$\begin{aligned} q & \sim \mathcal{N}(\beta_q^T v + \Sigma_q) \\ s & \sim \mathcal{N}(\beta_s^T v + \Sigma_s) \\ t & \sim \mathcal{N}(\beta_t^T v + \Sigma_t) \\ w & \sim \mathcal{N}(\beta_w^T v + \Sigma_w) \end{aligned} \tag{34}$$

We also assume that $v \in \mathbb{R}^r$ follows the normal distribution with known μ_v and Σ_v

$$v \sim \mathcal{N}(\mu_v, \Sigma_v). \tag{35}$$

We change the parameters as $n_1 = 100, n_2 = 100, m_1 = 100, m_2 = 100, M = \{10, 100, 1000\}, k = \{0.1 M, 0.5 M, 0.9 M\}, N = 10,000$. Each element of c, A was randomly generated from a uniform distribution $U(0, 1)$. Furthermore, b was set by generating a random solution x_0 and setting $b = A^{-1}x_0$. Each element of mu_v was randomly drawn from the uniform distribution $U(0, 5)$. Each element of Σ_v was randomly drawn from $U(0, 5)$ and made into a symmetric matrix by setting $\Sigma_v := (\Sigma_v + \Sigma_v^T)/2$. Each element of $\mu_q, \mu_s, \mu_t, \mu_w$ was randomly drawn from the uniform distribution $U(0, 1)$. Each element of $\Sigma_q, \Sigma_s, \Sigma_t, \Sigma_w$ was randomly drawn from $U(0, 1)$ and made into a symmetric matrix by the same method as Σ_v .

The result of the out-of-sample performance is summarized in Table 4. From Table 4, it can be seen that RO ($k = 1$), the proposed method, has the best out-of-sample performance, as in the case of small sized instance. We also find that the SAA and PP approaches have very poor out-of-sample performance. This result suggests that utilizing the auxiliary variable v and consideration of the prediction error will make the out-of-sample performance better.

The CPU Time to solve randomly generated samples is summarized in Table 5. From Table 5, SAA takes a long time when the sample size M is large. This is because it is necessary to solve the 2nd stage linear programming problem for each sample, i.e., M times. The PP is the fastest, regardless of sample size M . This is because PP solves 2nd stage linear programming problem only once regardless of the sample size. The PR was faster than the SAA and slower than the PP. This is because it is necessary to solve the 2nd stage linear programming problem for for each sample in the k -nearest neighborhood, i.e., k times. As $1 \leq k \leq M$, the relation of the CPU Time for these three approaches can be explained. Finally, the RO was faster than PR and slower than PP. This is because the proposed method solves only one SOCP regardless of the sample size M .

It is not possible to directly compare the speed with other papers because the assumption of the proposed framework is more complex. The closest one is that of Bertsimas and Van Parys (2017), in which their proposed algorithm was tested by the newsvendor problem with one decision variable and the portfolio allocation problem with the six decision

variables. In this study, the proposed method was tested to the two-stage problem with over 100 decision variables in each stage. These results indicate the proposed method has a better scalability compared to the existing alternative approaches.

Unfortunately, the result was not obtained within an hour for even larger data, e.g., $M \geq 10^4$ or $n_1 = n_2 \geq 10^3$. This is mainly because of the performance of the commercial solver. The robust counterpart derived in the proposed method is a SOCP and in theory can be solved efficiently. However, it is still nonlinear programming model and is difficult. Therefore, we need to consider developing the algorithm to exploit the special structure of the model for the future research.

Table 4. Average out-of-sample performance for randomly generated samples.

M	SAA	PP	PR	RO ($k = 0.1 M$)	RO ($k = 0.5 M$)	RO ($k = 0.9 M$)
10	11.09	0.53	0.53	0.35	4.56	1.19
100	12.30	13.36	0.39	0.22	0.72	1.70
1000	21.99	19.93	1.12	0.03	0.13	1.62

Table 5. Average CPU Time to solve randomly generated samples.

M	SAA	PP	PR	RO ($k = 0.1 M$)	RO ($k = 0.5 M$)	RO ($k = 0.9 M$)
10	0.014	0.001	0.007	0.027	0.059	0.169
100	0.257	0.002	0.593	0.037	0.027	0.071
1000	13.917	0.002	0.184	0.018	0.089	0.020

6. Conclusions

Business analytics has been more important than ever. In this field, the integration of predictive analytics and prescriptive analytics has enormous potential. However, existing studies applied them separately and thus ended up in the suboptimal solution.

In this study, we propose an alternative approach that integrates machine learning and robust optimization. The proposed method applied a non-parametric k -nearest neighbor prediction model given the observation of the auxiliary covariates. The enclosing minimum volume ellipsoid that contains k -nearest neighboring samples is applied to form the uncertainty set of uncertain parameters. The robust optimization is applied to minimize the worst-case objective function over the obtained uncertainty set.

The proposed algorithm utilizes a nonparametric prediction model and thus does not need to assume probability distribution. The proposed algorithm forms around k -nearest neighboring samples and thus has robustness against the prediction error. The proposed algorithm utilizes robust optimization over ellipsoidal uncertainty, for which the efficient algorithm has been extensively studied.

In the numerical experiment, we applied the proposed method to the two-stage linear predictive prescription problem. The proposed method outperforms the alternative approaches, in terms of the out-of-sample performance and the computation time.

For future research, we consider the connection with probability. This can be achieved by application of other nonparametric methods, such as kernel regression. These models have a connection to probability without assuming the probability distribution. We can draw a confidence region of the uncertain parameter, given the observation of the auxiliary variable. The minimum volume ellipsoid enclosing the training samples within the confidence region. By doing so, we can control the degree of conservatism. Another important issue is the development of a custom solver that can exploit the special structure of the problem. As the robust counterpart is the nonlinear SOCP, which can be difficult to solve when the problem size is large, this problem can be solved by modern convex optimization techniques for the large sized problem.

Funding: This work was supported by JSPS KAKENHI Grant Number 19K15243. This work was partly executed under the cooperation of organization between Waseda University and KIOXIA Corporation (former Toshiba Memory Corporation).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

- Davenport, T.H. Competing on analytics. *Harv. Bus. Rev.* **2006**, *84*, 98. [[PubMed](#)]
- Keenan, P.T.; Owen, J.H.; Schumacher, K. Introduction to Analytics. In *INFORMS Analytics Body of Knowledge*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2018; pp. 1–30.
- Lustig, I.; Dietrich, B.; Johnson, C.; Dziekan, C. The analytics journey. *Anal. Mag.* **2010**, *3*, 11–13.
- Evans, J.R.; Lindner, C.H. Business analytics: The next frontier for decision sciences. *Decis. Line* **2012**, *43*, 4–6.
- Ben-Tal, A.; El Ghaoui, L.; Nemirovski, A. *Robust Optimization*; Princeton University Press: Princeton, NJ, USA, 2009; Volume 28.
- Smith, J.E.; Winkler, R.L. The optimizer’s curse: Skepticism and postdecision surprise in decision analysis. *Manag. Sci.* **2006**, *52*, 311–322. [[CrossRef](#)]
- Bertsimas, D.; Kallus, N. From predictive to prescriptive analytics. *Manag. Sci.* **2020**, *66*, 1025–1044. [[CrossRef](#)]
- Bertsimas, D.; Gupta, V.; Kallus, N. Data-driven robust optimization. *Math. Program.* **2018**, *167*, 235–292. [[CrossRef](#)]
- Charnes, A.; Cooper, W.W. Chance-constrained programming. *Manag. Sci.* **1959**, *6*, 73–79. [[CrossRef](#)]
- Soyster, A.L. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Oper. Res.* **1973**, *21*, 1154–1157. [[CrossRef](#)]
- Ben-Tal, A.; Nemirovski, A. Robust convex optimization. *Math. Oper. Res.* **1998**, *23*, 769–805. [[CrossRef](#)]
- Ben-Tal, A.; Nemirovski, A. Robust solutions of uncertain linear programs. *Oper. Res. Lett.* **1999**, *25*, 1–13. [[CrossRef](#)]
- Ben-Tal, A.; Nemirovski, A. Robust solutions of linear programming problems contaminated with uncertain data. *Math. Program.* **2000**, *88*, 411–424. [[CrossRef](#)]
- El Ghaoui, L.; Lebret, H. Robust solutions to least-squares problems with uncertain data. *SIAM J. Matrix Anal. Appl.* **1997**, *18*, 1035–1064. [[CrossRef](#)]
- El Ghaoui, L.; Oustry, F.; Lebret, H. Robust solutions to uncertain semidefinite programs. *SIAM J. Optim.* **1998**, *9*, 33–52. [[CrossRef](#)]
- Bertsimas, D.; Sim, M. The price of robustness. *Oper. Res.* **2004**, *52*, 35–53. [[CrossRef](#)]
- Ben-Tal, A.; Nemirovski, A. Selected topics in robust convex optimization. *Math. Program.* **2008**, *112*, 125–158. [[CrossRef](#)]
- Gorissen, B.L.; Yanikoglu, İ.; den Hertog, D. A practical guide to robust optimization. *Omega* **2015**, *53*, 124–137. [[CrossRef](#)]
- Gabrel, V.; Murat, C.; Thiele, A. Recent advances in robust optimization: An overview. *Eur. J. Oper. Res.* **2014**, *235*, 471–483. [[CrossRef](#)]
- Sozuer, S.; Thiele, A.C. The state of robust optimization. In *Robustness Analysis in Decision Aiding, Optimization, and Analytics*; Springer: Cham, Switzerland, 2016; pp. 89–112.
- Delage, E.; Iancu, D.A. Robust multistage decision making. In *The Operations Research Revolution*; INFORMS: Catonsville, MD, USA, 2015; pp. 20–46.
- Delage, E.; Ye, Y. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Oper. Res.* **2010**, *58*, 595–612. [[CrossRef](#)]
- Ben-Tal, A.; Bhadra, S.; Bhattacharyya, C.; Nath, J.S. Chance constrained uncertain classification via robust optimization. *Math. Program.* **2011**, *127*, 145–173. [[CrossRef](#)]
- Dupacova, J.; Kopa, M. Robustness in stochastic programs with risk constraints. *Ann. Oper. Res.* **2012**, *200*, 55–74. [[CrossRef](#)]
- Xu, H.; Caramanis, C.; Mannor, S. A distributional interpretation of robust optimization. *Math. Oper. Res.* **2012**, *37*, 95–110. [[CrossRef](#)]
- Zymler, S.; Kuhn, D.; Rustem, B. Distributionally robust joint chance constraints with second-order moment information. *Math. Program.* **2013**, *137*, 167–198. [[CrossRef](#)]
- Wiesemann, W.; Kuhn, D.; Sim, M. Distributionally robust convex optimization. *Oper. Res.* **2014**, *62*, 1358–1376. [[CrossRef](#)]
- Ben-Tal, A.; Den Hertog, D.; De Waegenaere, A.; Melenberg, B.; Rennen, G. Robust solutions of optimization problems affected by uncertain probabilities. *Manag. Sci.* **2013**, *59*, 341–357. [[CrossRef](#)]
- Esfahani, P.M.; Kuhn, D. Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations. *Math. Program.* **2018**, *171*, 115–166. [[CrossRef](#)]
- Melin, P.; Castillo, O. A review on type-2 fuzzy logic applications in clustering, classification and pattern recognition. *Appl. Soft Comput.* **2014**, *21*, 568–577. [[CrossRef](#)]
- Pozna, C.; Precup, R.E. Applications of signatures to expert systems modelling. *Acta Polytech. Hung.* **2014**, *11*, 21–39.

32. Jammalamadaka, S.R.; Qiu, J.; Ning, N. Predicting a Stock Portfolio with the Multivariate Bayesian Structural Time Series Model: Do News or Emotions Matter? Available online: <http://www.ceser.in/ceserp/index.php/ijai/article/view/6255> (accessed on 6 January 2021).
33. Den Hertog, D.; Postek, K. *Bridging the Gap between Predictive and Prescriptive Analytics-New Optimization Methodology Needed*; Technical report; Tilburg University: Tilburg, The Netherlands, 2016.
34. Elmachoub, A.N.; Grigas, P. Smart “Predict, then Optimize”. *arXiv* **2017**, arXiv:1710.08005.
35. Larsen, E.; Lachapelle, S.; Bengio, Y.; Frejinger, E.; Lacoste-Julien, S.; Lodi, A. Predicting solution summaries to integer linear programs under imperfect information with machine learning. *arXiv* **2018**, arXiv:1807.11876.
36. Bertsimas, D.; Dunn, J.; Mundru, N. Optimal Prescriptive Trees. Available online: <https://pubsonline.informs.org/doi/10.1287/ijoo.2018.0005> (available on 16 April 2019.)
37. Dunn, J.W. Optimal Trees for Prediction and Prescription. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2018.
38. Bertsimas, D.; Van Parys, B. Bootstrap robust prescriptive analytics. *arXiv* **2017**, arXiv:1711.09974.
39. Yiannis, K.; Poulicos, P. Forecasting traffic flow conditions in an urban Network-comparison of multivariate and univariate approaches. *Transp. Res. Rec.* **2003**, *1857*, 74–84.
40. Boyd, S.; Boyd, S.P.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2004.