Bootstrap Robust Prescriptive Analytics

Dimitris Bertsimas*[1] and Bart Van Parys†[1]

[1] *Operations Research Center, Massachusetts Institute of Technology*

November 27, 2017

Abstract

We address the problem of prescribing an optimal decision in a framework where its cost depends on uncertain problem parameters Y that need to be learned from data. Earlier work by Bertsimas and Kallus (2014) transforms classical machine learning methods that merely predict Y from supervised training data $[(x_1, y_1), \ldots, (x_n, y_n)]$ into prescriptive methods taking optimal decisions specific to a particular covariate context $X = \bar{x}$. Their prescriptive methods factor in additional observed contextual information on a potentially large number of covariates $X = \bar{x}$ to take context specific actions $z(\bar{x})$ which are superior to any static decision z. Any naive use of limited training data may, however, lead to gullible decisions over-calibrated to one particular data set. In this paper, we borrow ideas from distributionally robust optimization and the statistical bootstrap of Efron (1982) to propose two novel prescriptive methods based on (nw) Nadaraya-Watson and (nn) nearest-neighbors learning which safeguard against overfitting and lead to improved out-of-sample performance. Both resulting robust prescriptive methods reduce to tractable convex optimization problems and enjoy a limited disappointment on bootstrap data. We illustrate the data-driven decision-making framework and our novel robustness notion on a small news vendor problem as well as a small portfolio allocation problem.

Keywords: Data Analytics, Distributionally Robust Optimization, Statistical Bootstrap, Nadaraya-Watson Learning, Nearest Neighbors Learning

1 Introduction

Traditionally, decision-making under uncertainty in operations research has largely focused on the classical stochastic optimization problem

$$z(\mathbb{Y}^\star) := z^\star \in \arg\min_z c(z, \mathbb{Y}^\star) := \mathrm{E}_{\mathbb{Y}^\star}\left[L(z, Y)\right] \tag{1}$$

and its multi-period generalizations. A wide spectrum of decision problems can indeed be cast as a stochastic optimization problem. Shapiro, Dentcheva, and Ruszczyńsk (2014) point out, for example, that problem (1) can be viewed as the first stage of a two-stage stochastic program, where the loss function L embodies the optimal value of a subordinate second-stage problem. Alternatively, problem (1) may also be interpreted as a generic learning problem in the spirit of statistical learning theory. Traditional stochastic optimization problems endeavor to find an optimal action $z^\star$ which among all feasible candidates in the set $\{z \in \mathtt{Z} \,:\, c(z, \mathbb{Y}^\star) < \infty\}$ has the lowest cost despite the presence of an uncertain parameter Y with distribution $\mathbb{Y}^\star$. Such parameters might include for instance historical demands in revenue management or price time series in portfolio optimization. Classical stochastic optimization problems try to capture the uncertain nature in which practical decisions inevitably need to be made through the distribution of a random variable Y perturbing the cost of an action z in an undeterministic way. We make the following standard assumption as to ensure that our classical stochastic optimization problem is well-posed.

*dbertsim@mit.edu
†vanparys@mit.edu

Assumption 1 (Convex Cost Model)**.** *We assume that the loss function $L(z, y)$ in $\mathtt{R}_+ \cup \{+\infty\}$ is convex and continuous in z for any $y \in \mathtt{Y}$ and a measurable function of y for any $z \in \mathtt{Z}$.*

By virtue of Assumption 1, finding an optimal action $z^\star$ reduces to solving a convex optimization problem. Unfortunately, Nemirovski and Shapiro (2006) have shown that stochastic optimization formulations tend to be computationally unattractive despite their convex nature for all but the simplest of problems. Their cost function is indeed characterized as a quadrature problem which are daunting in high-dimensional settings. Simply evaluating the cost which a fixed decision incurs is not at all trivial and indeed often a hard problem in its own regard. Even worse, an uncertain parametric influence described by a distribution is never observed directly in practice. Indeed, distributions are a product of modeling uncertainty rather than being a directly observable primitive. This makes classical stochastic optimization formulations not particularly well suited for modern decision-making.

Data, not distributions, should be the primitive for decision-making under uncertainty. Only data is ever observed directly and hence should be given the lead role in any decision-making problem. The primitive describing uncertainty is in this setting hence not a probability distribution but rather a collection of *training labels*

$$\mathtt{Y}_{n,t} := [y_1, \ldots, y_n] \tag{2}$$

on the uncertain parameters of direct interest. With the help of subscripts we make it explicit in this paper that only a limited number (n) of training (t) labels are observed. These training labels can then serve as the input to a statistical estimation method which aims to infer the distribution of the uncertain problem parameters out of the set of potential ones $\mathcal{Y}$. This inferred distribution then itself constitutes the input to an optimization problem that selects that action incurring the smallest predicted average loss. The problem is hence separated into a predictor component used to estimate the cost of decisions in light of the training labels $\mathtt{Y}_{n,t}$, and a prescriptor component which uses these predictions to select the most suitable course of action. It is evident that designing a predictor requires a statistical estimation perspective, while the subsequent search for an appropriate prescription is clearly more the task of an optimizer. Data-driven decision-making evidently requires a good marriage between these two components in order to function properly. From this perspective the popular sample average formulation does this job rather well by substituting the unknown distribution of the uncertain parameters Y with the empirical distribution $\mathbb{Y}_{n,t} := \frac{1}{n}\sum_{\mathtt{Y}_{n,t}} \delta_y$ of the training labels. Here, the δ_y denote the Dirac delta distributions on training labels.

Definition 1 (The Sample Average Formulation)**.** *The sample average formulation in the context of training labels $\mathtt{Y}_{n,t}$ with empirical distribution $\mathbb{Y}_{n,t}$ is given as*

$$z(\mathbb{Y}_{n,t}) = z_{n,t} \in \arg\min_z \, c(z, \mathbb{Y}_{n,t}). \tag{3}$$

This popular prescriptive method is treated here as an autonomous data-driven formulation of classical decision-making under uncertainty, rather than as an approximation attempt to stochastic optimization problems as done for instance in the excellent survey by Shapiro (2003). Although many interesting works deal with identifying conditions under which $z_{n,t} \to z^\star$ with an increasing amount n of data samples, we are not interested here in restating or discussing any such results. We want to treat data here as deterministic observations rather than random samples. To reflect this crucial change in perspective we refer to (3) as the sample average *formulation* rather than the sample average *approximation* which is its more common denomination.

Classical stochastic optimization is however unable to incorporate additional contextual information concerning observed covariates $X = \bar{x}$. Often indeed, covariate information such as weather forecasts, Twitter feeds, Google Trends data, ..., can be obtained before any decision needs to be made. One can wonder whether such auxiliary data contains any valuable information, however, as by definition none of the covariates X has any direct influence on the problem faced by the decision maker. While for instance price is of direct interest to a portfolio manager, Twitter chatter should indeed not be. A static decision maker should not assign any value to such auxiliary covariates. The situation changes dramatically if a particular covariate context $X = \bar{x}$ is revealed to the decision maker before any decision is made. An adaptive decision maker who can tailor the taken course of action to an observed context $\bar{x}$ may come to value the covariate context greatly.

The portfolio manager may for instance alter strategy when given prior notice about a relevant Twitter storm surrounding one of his assets. Despite not being immediately relevant to the stochastic optimization problem (1), such further context can have a dramatic indirect impact as it allows for superior more context specific decision-making. When decisions are to be made in a particular observed context $\bar{x}$, the problem in need of attention should not be the classical stochastic optimization problem (1) but rather

$$z(\bar{x}, \mathbb{M}^\star) := z^\star(\bar{x}) \in \arg\min_z c(z, \mathbb{Y}^\star(\bar{x})) := \mathrm{E}_{\mathbb{M}^\star}\left[L(z, Y)|X = \bar{x}\right]. \tag{4}$$

We denote problem (4) as a *prescriptive analytics* problem. The model distribution $\mathbb{M}^\star$ represents here the joint distribution between the labels and covariates. We use in this paper the notation $\mathbb{Y}^\star(\bar{x})$ as a shorthand for the distribution of the uncertain parameters Y conditioned on a particular observed context $\bar{x}$. Evidently, we have $z(\bar{x}, \mathbb{M}^\star) = z(\mathbb{Y}^\star(\bar{x}))$ providing a direct relation between stochastic optimization problems and data analytics problems. The taken decision $z^\star(\bar{x})$ hence adapts to the covariate context $\bar{x}$ observed allowing superior decision-making. It can be remarked that the analytics problem (4) generalizes the classical decision problem (1) as the former reduces to the latter when it so happens that the conditional distribution $\mathbb{Y}^\star(\bar{x}) = \mathbb{Y}^\star$ is context independent. In this case there is no point in taking adaptive decisions as $z^\star(\bar{x}) = z^\star$ and any covariate information can be ignored without incurring additional cost.

Classical prescriptive analytics problems are based on a hypothesized model $\mathbb{M}^\star$ in much the same way stochastic optimization problems were based on a hypothesized model $\mathbb{Y}^\star$. Consequently, neither are particularly suited for modern decision-making with data. Bertsimas and Kallus (2014) proceed to construct data-driven counterparts to the classical prescriptive analytics problem (4) with the help of nonparametric predictive learning methods applied to the *supervised training data*

$$\mathtt{M}_{n,t} := [m_1 = (x_1, y_1), \ldots, m_n = (x_n, y_n)]. \tag{5}$$

The labels $\mathtt{Y}_{n,t}$ may only represent a tiny fraction of the available data $\mathtt{M}_{n,t}$. In today's data-rich world this auxiliary data on associated covariates X such as weather forecasts, Twitter feeds, financial market data, Yelp reviews, ..., may in fact represent the bulk of all available data. Nonparametric predictive learning methods treat data as if it where n independent samples from a common but otherwise completely unknown model distribution on the joint event space $\mathtt{M} := \mathtt{X} \times \mathtt{Y}$ between covariates and labels. Their minimal assumptions posed on the data makes them a versatile tool in the learning which model bets fits the data. The set of potential statistical models $\mathcal{M}$ consists here of all probability distributions on the event space $\mathtt{M}$. In honor of the sample average formulation, we will denote the prescriptive methods of Bertsimas and Kallus (2014) as supervised learning formulations. The empirical model $\mathbb{M}_{n,t} := \frac{1}{n}\sum_{\mathtt{M}_{n,t}} \delta_{(x,y)}$ will serve the same role as the empirical distribution in the sample average formulation. Their supervised learning formulations are distinct only in the way they use the empirical model to predict the distribution of Y in the covariate context $\bar{x}$.

Definition 2 (Predictive Learner (Bertsimas and Kallus, 2014))**.** *A predictive learner is a function* $\mathbb{Y}_n^\ell : \mathtt{X} \times \mathcal{M} \to \mathcal{Y}$ *mapping a model in* $\mathcal{M}$ *to a distribution in* $\mathcal{Y}$ *for any particular covariate context in* $\mathtt{X}$*.*

Such predictive learners address what has been called the conditional density estimation problem in the statistics community. It is hoped indeed that $\mathbb{Y}_n^\ell(\bar{x}, \mathbb{M}_{n,t}) \to \mathbb{Y}^\star(\bar{x})$ with an increasingly large amount of data. One particularly naive contextual learner can be found as the empirical conditional probability distribution

$$\mathbb{Y}_n^\star(\bar{x}, \mathbb{M}_{n,t}) := \textstyle\sum_{\{(x=\bar{x},y)\in\mathtt{M}_{n,t}\}} \delta_y / |\{(x = \bar{x}, y) \in \mathtt{M}_{n,t}\}|.$$

We remark, however, that the training model distribution $\mathbb{M}_{n,t}$ is discrete and supported on a set $\mathtt{M}_n$ counting at most n points. The set of all such models supported by the training data is denoted as $\mathcal{M}_n$. Evidently, this naive contextual learner ends up disregarding all data points in the training data set whose auxiliary data $x \neq \bar{x}$. Consequently, it is only properly defined for $\bar{x}$ in the covariate support set $\mathtt{X}_n$. We hence clearly have a need for more sensible alternatives.

We defer the discussion of the ways in which the predictive learners $\mathbb{Y}_n^\ell$ proposed by Bertsimas and Kallus (2014) are distinct to the next section. Instead, we first discuss here what they have in common. Bertsimas and Kallus (2014) base their supervised learning formulations directly on a predictive learner as in

Definition 2. Given indeed an estimate $\mathbb{Y}^{\ell}_{n,t}(\bar{x}) := \mathbb{Y}^{\ell}_{n}(\bar{x}, \mathbb{M}_{n,t})$ of the distribution of the uncertain problem parameters conditional on the particular covariate context of interest, it is indeed sensible to prescribe the action $z(\mathbb{Y}^{\ell}_{n,t}(\bar{x}))$.

Definition 3 (Supervised learning formulation (Bertsimas and Kallus, 2014))**.** *A supervised learning formulation at a given context $\bar{x}$ with training data $\mathtt{M}_{n,t}$ is defined as*

$$z^{\ell}_{n}(\bar{x}, \mathbb{M}_{n,t}) := z^{\ell}_{n,t}(\bar{x}) \in \arg\min_{z} \; c^{\ell}_{n}(z, \mathbb{M}_{n,t}, \bar{x}) := c(z, \mathbb{Y}^{\ell}_{n,t}(\bar{x})). \tag{6}$$

Any data driven method however needs to be concerned with the confidence it can place in decisions tailored to a specific training data set. It is clear that when given only a limited amount of data, any method must be guarded against overfitting phenomena. If we take a decision $z^{\ell}_{n,t}(\bar{x})$ calibrated to one particular training data set $\mathtt{M}_{n,t}$ and evaluate its performance on a test data set

$$\mathtt{M}_{n,r} := [M_{1,r} = (X_{1,r}, Y_{1,r}) \sim \mathbb{M}^{\star}, \ldots, M_{n,r} = (X_{n,r}, Y_{n,r}) \sim \mathbb{M}^{\star}], \tag{7}$$

even when generated by the same model $\mathbb{M}^{\star}$ (Michaud, 1989), then the resulting test performance is often disappointing. With the help of subscripts we make it explicit here that $\mathtt{M}_{n,r}$ consists of a limited number (n) of data points originating (r) from the same model as the training data. The odds are indeed that the originally proposed prescription $z^{\ell}_{n,t}(\bar{x})$ will cost more as validated on test date than originally budgeted based on the training data. That is, the probability

$$\mathbb{M}^{\star n}\Big(\overbrace{c(z^{\ell}_{n,t}(\bar{x}), \mathbb{Y}^{\ell}_{n}(\bar{x}, \mathbb{M}_{n,r}))}^{\text{out-of-sample cost}} > \overbrace{c^{r,\ell}_{n,t} := c^{\ell}_{n}(z^{\ell}_{n,t}(\bar{x}), \mathbb{M}_{n,t}, \bar{x})}^{\text{training cost}} \Big) \tag{8}$$

of being disappointed out of sample may be unacceptably high. In the previous equation, we denoted $\mathbb{M}_{n,r} = \frac{1}{n}\sum_{\mathtt{M}_{n,r}} \delta_{(x,y)}$ as the empirical model fitted to the test data $\mathtt{M}_{n,r}$. This clearly adverse phenomena is well known in the literature as the optimizer's curse or overfitting. The prescriptive analytics framework as presented by Bertsimas and Kallus (2014) does not have any inherent defense mechanism to such adversarial effects. When working with data instead of distributions, however, one should safeguard against solutions which display promising training performance, but lead to out-of-sample disappointment. It is standard practice in machine learning to guard against overfitting by requiring that predictions do not disappoint on a large fraction of out-of-sample data. That is, by designing robust supervised learning formulations for which the probability (8) of being left disappointed is bounded above by a small quantity $b \in [0, 1)$.

Contributions

Notice that the previous robustness notion requiring (8) to be small hinges on our ability to obtain the test data $\mathtt{M}_{n,r}$. However, it is clear that obtaining such test data in practice is not a viable approach. Simply generating more data is in general not possible. We would also like to prevent overfitting without appealing to any statistical assumptions on how the training data was generated. The training data may not have been generated by any statistical model $\mathbb{M}^{\star}$ at all, preventing test data such $\mathtt{M}_{n,r}$ to be properly defined in the first place. In practice overfitting is indeed prevented not by making exotic assumptions on the data generating process but rather through cross validation or bootstrapping. This latter technique considers bootstrap data

$$\mathtt{M}_{n,b} := [M_{1,b} = (X_{1,b}, Y_{1,b}) \sim \mathbb{M}_{n,t}, \ldots, M_{n,b} = (X_{n,b}, Y_{n,b}) \sim \mathbb{M}_{n,t}], \tag{9}$$

generated independently from resampling the original training data and enforces robustness by requiring decisions to do well on a large fraction of such random bootstrap data. In a departure from previous work on robustness with data, we will present here prescriptive methods which are robust in exactly this sense. That is, a bootstrap robust formulation only disappoints with respect to its nominal counterpart on a small fraction b of the random bootstrap data $\mathtt{M}_{n,b}$.

Definition 4 (Bootstrap Robustness)**.** *The robust cost $c^{\mathrm{r},\ell}_{n} : \mathtt{Z} \times \mathcal{M}_n \times \mathtt{X} \to \mathtt{R}$ together with its associated prescriptor $z^{\mathrm{r},\ell}_{n} : \mathcal{M}_n \times \mathtt{X} \to \mathtt{R}$ are said to suffer bootstrap disappointment $b \in [0, 1)$ if we have the inequality*

$$\mathbb{M}^{n}_{n,t}\Big(\overbrace{c(z^{\mathrm{r},\ell}_{n,t}(\bar{x}), \mathbb{Y}^{\ell}_{n}(\bar{x}, \mathbb{M}_{n,b}))}^{\text{nominal bootstrap cost}} > \overbrace{c^{r,\ell}_{n,t} := c^{\mathrm{r},\ell}_{n}(z^{\mathrm{r},\ell}_{n,t}(\bar{x}), \mathbb{M}_{n,t}, \bar{x})}^{\text{robust training cost}} \Big) \leq b. \tag{10}$$

We defer here the discussion of the merits and limitations of our novel robustness notion to Section 4. We consider supervised learning formulations based on two classical nonparametric predictive learning methods: (nw) Nadaraya-Watson learning and (nn) nearest-neighbors learning. Bertsimas and Kallus, 2014 transformed both these predictive learning methods into a distinct supervised learning formulation turning supervised data $\mathtt{M}_{n,t}$ in a decision adapted to a given context $\bar{x}$. These supervised learning formulations should be interpreted as a direct counterpart to the sample average formulation translated to the supervised data setting. Similar to the sample average formulation, that is, they have no inherent defense mechanism against over-calibration to the particular data set $\mathtt{M}_{n,t}$.

The main technical innovations presented in this work are:

1. Definition 4 advances a novel notion of robustness based directly on the statistical bootstrap by Efron, 1982. Although our robustness notion does not yield statistical consistency guarantees similar to those found in Shapiro (2003), it presents a more data-centered alternative perspective. Indeed, our bootstrap robustness guarantee can be verified without assuming the data to be statistical at all. Instead, we treat the data for what it is: deterministic observations. We believe this to be a more appropriate point of view as the statistical alternative is often not falsifiable and needs to be taken on faith rather than on evidence.

2. We make both the Nadaraya-Watson and nearest neighbors formulation advanced by Bertsimas and Kallus (2014) resilient against the adverse effects of overfitting by formulating a distributionally robust counterpart. We indicate that the resulting robust supervised learning formulations are computationally as tractable as their nominal counterparts. When for instance the nominal supervised learning formulation reduces to a tractable convex optimization problem then so will its robust counterpart. The previous crucial observation makes both of our robust formulations practically viable.

3. One particular robust counterpart based on the relative entropy (Kullback and Leibler, 1951) distance is proven to safeguard against bootstrap overfitting as stated in Definition 4. We derive practical finite sample bootstrap performance guarantees as in (10) regarding the resulting robust supervised learning formulation. For this particular bootstrap robust counterpart we derive a more explicit tractable reformulation based on convex duality.

Finally, we present the efficacy of our three proposed data-driven formulations on a small news vendor problem as well as a small portfolio allocation problem. We published a `Julia` implementation of the ideas and examples in this work on `Github` at https://github.com/vanparys/bootstrap-robust-analytics-julia.

Paper Outline

In Section 2, we formally introduce the Nadaraya-Watson and nearest neighbors learning formulations as first introduced by Bertsimas and Kallus (2014). In Section 3, we present a generic robust counterpart to these nominal supervised learning formulations based on our concept of a model distance function. In the same section we show that when these model distance functions are convex, the associated generic robust formulation is as tractable as the original nominal counterpart. In Section 4, we tie generic robust supervised learning formulations together with guaranteed bootstrap sample performance by singling out a particular convex model distance function based on the relative entropy distance by Kullback and Leibler (1951). For this bootstrap distance function we derive additional more efficient reformulations of the associated robust supervised learning formulations through convex duality. In Section 5, finally, we illustrate our data-driven decision-making framework and robustness notion on a small news vendor problem as well as a small portfolio allocation problem.

2 Nominal Prescriptive Analytics

The supervised learning formulations of Definition 3 are distinct only in so far they are based on a different predictive learner. We will focus on supervised learning formulations based on two popular nonparametric

learning methods (nw) Nadaraya-Watson learning, and (nn) nearest neighbors learning. Both these supervised learning formulations present a distinct approach to decision-making based on the supervised training data $\mathtt{M}_{n,t}$. This section briefly introduces the learning methods advanced by Bertsimas and Kallus, 2014 and as such contains no new results but rather introduces additional notation. Depending on the particular learning method employed, the same supervised training data is treated differently leading to a distinct prescribed course of action. Both Nadaraya-Watson and nearest neighbors are local memory-based learning methods which require little to no training effort (Friedman, Hastie, and Tibshirani, 2001); the lion share of the work will get done at evaluation time. There is a vast literature on local learning methods which we will not attempt to summarize. We refer to Scott (2015) and Silverman (1986) for extensive bibliographies. We describe here in detail in what sense the predictive Nadaraya-Watson learner $\mathbb{Y}_n^{\mathrm{nw}}$ and the predictive nearest neighbors learner $\mathbb{Y}_n^{\mathrm{nn}}$ differ and what they have in common. Contrary to the naive contextual learner $\mathbb{Y}_n^\star$ introduced earlier, they will be properly defined outside of the covariate training support $\mathtt{X}_n$ as well.

2.1 The Nadaraya-Watson Formulation

Definition 5 (Nadaraya-Watson Learning (Nadaraya, 1964))**.** *The Nadaraya-Watson learner* $\mathbb{Y}_n^{\mathrm{nw}} : \mathtt{X} \times \mathcal{M}_n \to \mathcal{Y}$ *contextualizes an empirical model* $\mathbb{M}_n$ *on the context* $\bar{x}$ *using*

$$\mathbb{Y}_n^{\mathrm{nw}}(\bar{x}, \mathbb{M}_n) := s \cdot \textstyle\sum_{\mathtt{M}_n} S_n(x - \bar{x}) \cdot \mathbb{M}_n(x, y) \cdot \delta_y, \tag{11a}$$

with normalization factor $s > 0$ *implicitly given as the solution to*

$$1 = s \cdot \textstyle\sum_{\mathtt{M}_n} S_n(x - \bar{x}) \cdot \mathbb{M}_n(x, y), \tag{11b}$$

where $S_n(\Delta x) := S(\Delta x / h_n)$ *using a given smoother function* $S : \mathtt{X} \to \mathtt{R}_+$.

The Nadaraya-Watson formulation will be based on the previous predictive learner. The Nadaraya-Watson contextual training model $\mathbb{Y}_{n,t}^{\mathrm{nw}}(\bar{x}) := \mathbb{Y}_n^{\mathrm{nw}}(\bar{x}, \mathbb{M}_{n,t})$ shares its support $\mathtt{Y}_n$ with the empirical distribution $\mathbb{Y}_{n,t}$ of the label data. Nevertheless, the learned contextual model weighs the data samples relative to each other using a smoother function S and bandwidth parameter h_n as opposed to simply inversely proportional to the number of samples. The normalization factor s is included to let the sum of all these weights add up to one assuring that the contextual model is in fact a probability distribution. Some common popular choices of smoothers are given in Figure 1. The Epanechnikov (1969) smoother is optimal in an asymptotic mean square error sense, though the loss of efficiency is small for the smoother functions listed previously. For the theoretical results in this paper, the particular smoother function employed will not be of great consequence. That being said, the choice of smoother may have a significant practical impact on the performance of the Nadaraya-Watson learner and must be chosen carefully based on the training data at hand. Often, cross-validation comes to mind in practice.

Nadaraya-Watson learners are particularly amenable to theoretical analysis due mostly to their simplicity. The Nadaraya-Watson learner can indeed be shown to be consistent, i.e., $\mathbb{Y}_{n,t}^{\mathrm{nw}}(\bar{x}) \to \mathbb{Y}^\star(\bar{x})$, when using an appropriately scaled bandwidth parameter h_n for any of the smoother functions introduced in Figure 1. We refer for a more rigorous discussion to the work of Nadaraya (1964) and Watson (1964). Although the particular choice of the bandwidth parameter is not import for consistency of the associated learning formulation, it is nevertheless crucial for practical performance. Li and Racine (2007) give a decent rule of thumb to appropriately select the bandwidth parameter as $h_n \approx \sigma_{x,t} \cdot n^{-1/(\dim \mathtt{X}+1)}$ where $\sigma_{x,t}$ is the empirical standard deviation of the marginal $\mathbb{X}_{n,t} := \frac{1}{n} \sum_{\mathtt{X}_{n,t}} \delta_x$ and $\dim \mathtt{X}$ the dimension of the auxiliary data. Choosing the bandwidth based on cross-validation usually results though in better performance compared to such analytical formulas. The final Nadaraya-Watson formulation uses the learned contextual model to take that course of action with the smallest predicted cost.

Definition 6 (The Nadaraya-Watson Formulation (Bertsimas and Kallus, 2014))**.** *The Nadaraya-Watson formulation at a given context* $\bar{x}$ *is defined as*

$$z_n^{\mathrm{nw}}(\bar{x}, \mathbb{M}_{n,t}) := z_{n,t}^{\mathrm{nw}}(\bar{x}) \in \arg\min_z c_n^{\mathrm{nw}}(z, \mathbb{M}_{n,t}, \bar{x}) := c(z, \mathbb{Y}_{n,t}^{\mathrm{nw}}(\bar{x}) := \mathbb{Y}_n^{\mathrm{nw}}(\bar{x}, \mathbb{M}_{n,t})). \tag{12}$$

Name	Smoother S
Uniform	$\frac{1}{2}\mathbb{1}_{\|\Delta x\|\leq 1}$
Epanechnikov	$\frac{3}{4}(1-\|\Delta x\|^2)\mathbb{1}_{\|\Delta x\|\leq 1}$
Tricubic	$\frac{70}{81}(1-\|\Delta x\|^3)^3\mathbb{1}_{\|\Delta x\|\leq 1}$
Gaussian	$\exp(-\|\Delta x\|^2/2)/\sqrt{2\pi}$

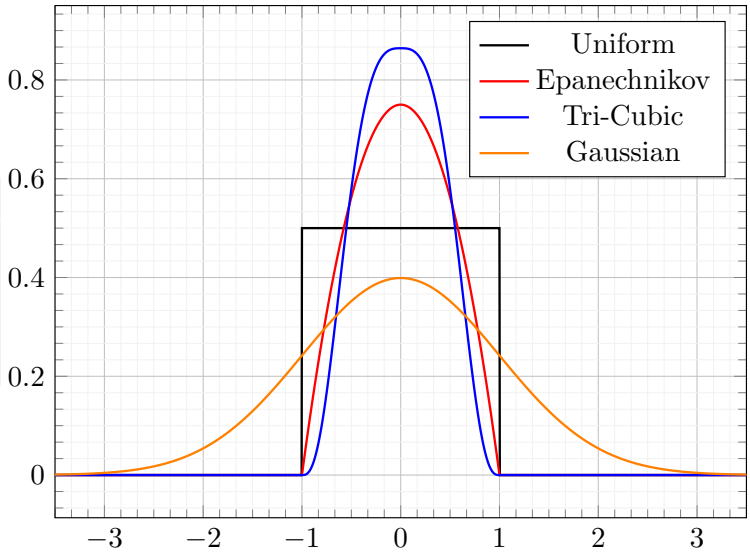


Figure 1: A comparison of popular common smoother functions S. The tricubic smoother has compact support and has two continuous derivatives at the boundary of its support, while the Epanechnikov smoother has none. The Gaussian smoother is continuously differentiable, but has infinite support.

2.2 The Nearest Neighbors Formulation

The nearest neighbor learning formulation only differs from the Nadaraya-Watson formulation in what contextual learner it employs. The nearest neighbors formulation considers, as its name suggests, the closely related nonparametric nearest neighbors learner (Altman, 1992). The nearest neighbors contextual learner considers only the smallest neighborhood around its context of interest $\bar{x}$ containing no less than k_n observations and will simply ignore any other data completely. Nearest neighbors learning is one of the most fundamental yet very simple learning methods and is discussed in virtually any textbook on machine learning. It is a common choice for learning when there is a lot of data but little or no prior knowledge about the distribution of that data.

A neighborhood implies a metric, and hence we assume first that we are given a function quantifying the proximity between any data point and the context of interest. Although this particular distance function may have a big influence on the quality of nearest neighbors learning, it will be inconsequential for our theoretical results. In fact, we do not even need the distance function to be a metric at all. What we do assume though is that the distance function allows us to order distinct data points on proximity to the covariate context of interest in a unique way. This can be achieved for instance by equipping a standard metric with a tie breaking rule. The resulting distance function can rely on the entire data sample and not only on the covariate marginal. Hence, we allow the distance function $d : \mathtt{M} \times \mathtt{X} \to \mathtt{R}_+$ to be a function of the entire data sample enjoying a discrimination property, i.e., $d(m, \bar{x}) = d(m', \bar{x}) \implies m = m'$. Please note that our definition of nearest neighbors which we will put forward shortly hereafter must take into account that the same data point can be observed multiple times in a training data set.

To accommodate this last technical issue, we must first introduce some concepts core to nearest neighbors learning. Let us define first what is meant with neighborhoods in the support $\mathtt{M}_n$ of the training data around a given context of interest $\bar{x}$. We divide the support $\mathtt{M}_n$ in increasingly large nested neighborhood sets

$$\mathtt{N}_n(\bar{x}, j) := \left\{m \in \mathtt{M}_n \ : \ d(m, \bar{x}) \leq R^\star_{n,j}\right\} \quad \text{with} \quad R^\star_{n,j} := \inf\left\{R \geq 0 \ : \ |\{m \in \mathtt{M}_n \ : \ d(m, \bar{x}) \leq R\}| \geq j\right\}$$

each containing those j points in the support $\mathtt{M}_n$ closest to our context of interest $\bar{x}$. The neighborhood sets $\{\mathtt{N}_n(\bar{x}, j) : j \in \{0, \ldots, |\mathtt{M}_n|\}\}$ are uniquely defined thanks to the discrimination property of the distance function considered. We will take here the zeroth neighborhood set $\mathtt{N}_{n,t}(\bar{x}, 0)$ to mean the empty set while evidently the $|\mathtt{M}_n|$-th neighborhood $\mathtt{N}_n(\bar{x}, |\mathtt{M}_n|)$ coincides with the entire support $\mathtt{M}_n$. The nearest neighbors

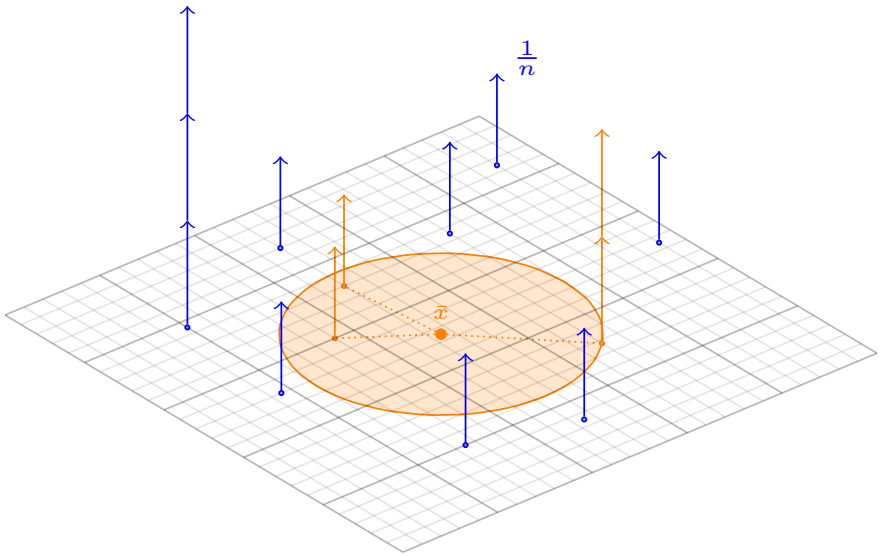


Figure 2: The three and four nearest neighbors (in orange) of the context of interest $\bar{x}$. We depict the neighborhood set $\mathtt{N}_n(\bar{x}, 3)$ as the orange circles in the support set $\mathtt{M}_n$. This neighborhood contains both the three and four nearest neighbors around $\bar{x}$ as the most distant nearest neighbor was seen twice in the training data. The orange circle visualizes the metric d implicit in the concept of nearest neighbors learning.

formulation will use the following predictive learner.

Definition 7 (Nearest Neighbors Learner)**.** *The k_n nearest neighbors learner $\mathbb{Y}_n^{\mathrm{nn}} : \mathtt{X} \times \mathcal{M}_n \to \mathcal{Y}$ contextualizes an empirical model $\mathbb{M}_n$ in the context $\bar{x}$ using*

$$\mathbb{Y}_n^{\mathrm{nn}}(\bar{x}, \mathbb{M}_n) := s \cdot \textstyle\sum_{\mathtt{N}_n(\bar{x},j)} S_n(x - \bar{x}) \cdot \mathbb{M}_n(x, y) \cdot \delta_y, \tag{13a}$$

with normalization factor $s > 0$ and neighborhood parameter $j \in [1, \ldots, |\mathtt{M}_n|]$ implicitly defined as

$$1 = s \cdot \textstyle\sum_{\mathtt{N}_n(\bar{x},j)} S_n(x - \bar{x}) \cdot \mathbb{M}_n(x, y), \tag{13b}$$

$$\mathbb{M}_n \in \mathcal{N}_n(\bar{x}, j) := \left\{ \mathbb{M} \in \mathcal{M}_n \;:\; \tfrac{k_n}{n} \leq \textstyle\sum_{\mathtt{N}_n(\bar{x},j)} \mathbb{M}(x, y),\ \tfrac{k_n - 1}{n} \geq \sum_{\mathtt{N}_n(\bar{x},j-1)} \mathbb{M}(x, y) \right\}, \tag{13c}$$

where once again $S_n(\Delta x) := S(\Delta x / h_n)$ using a smoother $S : \mathtt{X} \to \mathtt{R}_+$.

The nearest neighbors learner is akin to the Nadaraya-Watson learner but is blind to data outside of its neighborhood of interest. This neighborhood of interest corresponds to the smallest neighborhood set $\mathtt{N}_n(\bar{x}, j)$ containing at least k_n neighboring data points of the training data set. The set $\mathcal{N}_n(\bar{x}, j)$ indeed contains all empirical models $\mathbb{M}_n$ corresponding to data counting at least k_n neighbors in the j-th neighborhood $\mathtt{N}_n(\bar{x}, j)$ while not more than $k_n - 1$ neighbors in any smaller neighborhood around $\bar{x}$. As the neighborhoods around any context of interest $\bar{x}$ are nested, the last condition needs only to be enforced with regards to the $(j-1)$-th neighborhood $\mathtt{N}_n(\bar{x}, j-1)$ as done in (13c). The normalization factor s is again included to assure the contextual model is a probability distribution. The smoother function S determines the importance of each observations within the neighborhood of interest to the final contextual model $\mathbb{Y}_{n,t}^{\mathrm{nn}}(\bar{x}) := \mathbb{Y}_n^{\mathrm{nn}}(\bar{x}, \mathbb{M}_{n,t})$ of the training data $\mathtt{M}_{n,t}$.

The classical nearest neighbors learner takes the empirical mean of the labels of the k_n nearest neighbors of $\bar{x}$ as its best prediction. The classical nearest neighbor learner hence corresponds to its weighted counterpart (13) using the naive smoother $S(x) = 1$. As pointed out by Friedman, Hastie, and Tibshirani (2001, Section 2.8.2), this particular learner can be interpreted too as a Nadaraya-Watson learner using the uniform smoother function and a data dependent bandwidth parameter h_n. The classical nearest neighbors learner is

consistent, i.e., $\mathbb{Y}^{\text{nn}}_{n,t}(\bar{x}) \to \mathbb{Y}^{\star}(\bar{x})$, provided that the number of neighbors k_n and the bandwidth parameter h_n considered are scaled appropriately with the number of training data samples. We refer for a more rigorous discussion on consistency to the work of Altman (1992). One particular appropriate scaling which is often used as rule of thumb concerning the necessary number of nearest neighbors is $k_n \approx \sqrt{n}$. The bandwidth parameter can be scaled as was done in case of the Nadaraya-Watson formulation. It can indeed be remarked that although the particular choice of the hyper parameters such as the distance metric d employed, the smoother S, the number of nearest neighbors k_n, and the bandwidth parameter h_n is not important for consistency of the associated contextual learner, it is nevertheless crucial for practical performance. The final nearest neighbors formulation uses the learned nearest neighbors model to take that course of action with the smallest predicted cost.

Definition 8 (The Nearest Neighbors Formulation (Bertsimas and Kallus, 2014))**.** *The nearest neighbors formulation at a given context $\bar{x}$ is defined as*

$$z^{\text{nn}}_n(\bar{x}, \mathbb{M}_{n,t}) := z^{\text{nn}}_{n,t}(\bar{x}) \in \arg\min_z \; c^{\text{nn}}_n(z, \mathbb{M}_{n,t}, \bar{x}) := c(z, \mathbb{Y}^{\text{nn}}_{n,t}(\bar{x}) := \mathbb{Y}^{\text{nn}}_n(\bar{x}, \mathbb{M}_{n,t})). \tag{14}$$

3 Robust Prescriptive Analytics

When working with data instead of models, one should safeguard against making decisions which display promising training performance, but lead to out-of-sample disappointment. The nominal supervised learning formulations discussed before are indeed gullible and tend to be over-calibrated to one particular data set. It is clear that when given only a limited amount of training data $\mathtt{M}_{n,t}$, any data-driven method must be guarded against such overfitting phenomena.

Distributionally robust optimization has attracted significant attention as it provides the sample average formulation with a disciplined safeguard mechanism against overfitting. By using a robust counterpart with respect to an ambiguity set of distributions around an estimated nominal one, they were shown by Van Parys, Esfahani, and Kuhn (2017) to be minimally biased while still enjoying statistical out-of-sample guarantees. Many interesting choices of the ambiguity set furthermore result in a tractable overall decision-making approach. The ambiguity set can be defined, for example, through confidence intervals for the distribution's moments as done by Delage and Ye (2010), Stellato, Van Parys, and Goulart (2017), Van Parys, Goulart, and Kuhn (2016), and Van Parys, Goulart, and Morari (2015). Alternatively, Wang, Glynn, and Ye (2016) use an ambiguity set that contains all distributions that achieve a prescribed level of likelihood, while Bertsimas, Gupta, and Kallus (2014) based theirs on models which pass a statistical hypothesis test. Distance-based ambiguity sets contain all models sufficiently close to a reference with respect to probability metrics such as the Prokhorov metric (Erdoğan and Iyengar, 2006), the Wasserstein distance (Mohajerin Esfahani and Kuhn, 2015; Pflug and Wozabal, 2007), or the total variation distance (Sun and Xu, 2016).

In this paper, we generalize distributionally robust optimization to supervised learning formulations as well. We construct generic robust supervised learning formulations with the help of a model distance function. The resulting robust supervised learning formulations should suffer only a limited out-of-sample disappointment (10) on data $\mathtt{M}_{n,b}$ generated using the statistical bootstrap. Generic robust supervised learning formulations are not necessarily robust in the sense put forward in Definition 4. In the next section we will show that such bootstrap robustness guarantees can be obtained by considering a very particular bootstrap distance function. However, we will concern ourselves in this section only with showing the practical viability of generic robust supervised learning formulations with respect to any model distance function.

Definition 9 (Model Distance Function)**.** *A model distance function $D : \mathcal{M}_n \times \mathcal{M}_n \to \mathtt{R}_+$ is a function quantifying the distance between two empirical models enjoying the following property:*

(i) Discrimination: $D(\mathbb{M}, \mathbb{M}') \geq 0$ *for all* $\mathbb{M}$ *and* $\mathbb{M}'$*, while* $D(\mathbb{M}', \mathbb{M}) = 0$ *if and only if* $\mathbb{M}' = \mathbb{M}$.

(ii) Convexity: $D(\mathbb{M}, \mathbb{M}')$ *is a convex function of* $\mathbb{M}$ *in* $\mathcal{M}$ *for all fixed* $\mathbb{M}'$.

We define first a generic robust counterpart to a nominal supervised learning formulation with respect to the ambiguity set $\{\mathbb{M} \;:\; D(\mathbb{M}, \mathbb{M}_{n,t}) \leq r_n\}$ consisting of all empirical models at distance not exceeding r_n.

Type	Formulation $D(\mathbb{M}, \mathbb{M}') =$
Pearson (χ^2)	$\sum_{\mathtt{M}_n} \frac{(\mathbb{M}(x,y) - \mathbb{M}'(x,y))^2}{\mathbb{M}'(x,y)}$
Entropy	$\sum_{\mathtt{M}_n} \log\left(\frac{\mathbb{M}(x,y)}{\mathbb{M}'(x,y)}\right) \mathbb{M}(x,y)$
Burg Entropy	$\sum_{\mathtt{M}_n} \log\left(\frac{\mathbb{M}'(x,y)}{\mathbb{M}(x,y)}\right) \mathbb{M}'(x,y)$
f-Divergence	$\sum_{\mathtt{M}_n} f\left(\frac{\mathbb{M}(x,y)}{\mathbb{M}'(x,y)}\right) \mathbb{M}'(x,y)$
Wasserstein	$\min_{T:\mathtt{M}_n \times \mathtt{M}_n \to \mathtt{R}_+} \left\{ \sum_{\mathtt{M}_n \times \mathtt{M}_n} T(m,m') \cdot \|m - m'\|_2^2 \;:\; \sum_{\mathtt{M}_n} T(\circ, m') = \mathbb{M}(\circ),\; \sum_{\mathtt{M}_n} T(m, \circ) = \mathbb{M}'(\circ) \right\}$

Table 1: Model distance functions based on popular probability divergence metrices. The f-divergences give rise to a model distance function for convex functions f with $f(1) = 0$. The Pearson and Burg entropy are particular cases for $f(t) = t^2 - 1$ and $f(t) = -\log(t)$, respectively. The Wasserstein distance is defined with the help of a linear optimization problem over a transport map T of dimension $|\mathtt{M}_n| \times |\mathtt{M}_n|$. Postek, den Hertog, and Melenberg (2016) provide and discus many more probability divergences in great detail.

Definition 10 (Robust Supervised learning Formulations)**.** *A robust supervised learning formulation with respect to the model distance function D at a given context $\bar{x}$ with training data $\mathtt{M}_{n,t}$ is defined as*

$$z_n^{\mathrm{r},\ell}(\bar{x}, \mathbb{M}_{n,t}) := z_{n,t}^{\mathrm{r},\ell}(\bar{x}) \in \arg\min_z \; c_n^{\mathrm{r},\ell}(z, \mathbb{M}_{n,t}, \bar{x}) := \sup \quad c(z, \mathbb{Y}_n^\ell(\bar{x}, \mathbb{M})) \\ \text{s.t.} \quad \mathbb{M} \in \mathcal{M}, \\ D(\mathbb{M}, \mathbb{M}_{n,t}) \leq r_n. \tag{15}$$

Due to the discrimination property of the model distance function, the nominal supervised learning formulation is recovered when the robustness radius tends towards zero. In that case we are indeed merely robust with respect to the singleton $\{\mathbb{M} \in \mathcal{M} \;:\; D(\mathbb{M}, \mathbb{M}_{n,t}) \leq 0\} = \{\mathbb{M}_{n,t}\}$. Using a robust counterpart instead of nominal supervised learning formulations will help us protect against making prescriptions which do well on the training data set but tend to disappoint on unseen data. The robust training prescription $z_{n,t}^{\mathrm{r},\ell}(\bar{x})$ indeed does well not on one particular training model $\mathbb{M}_{n,t}$ but on all models $\{\mathbb{M} \;:\; D(\mathbb{M}, \mathbb{M}_{n,t}) \leq r_n\}$ at distance less than r_n simultaneously. The particular distance function D dictates which distributions are close to the nominal training model and consequently should be chosen with care. Several popular choices are listed in Table 1. In the next section, we will single out one particularly relevant model distance function in the context of the bootstrap disappointment defined in (10).

As mentioned earlier, the final obtained robust supervised learning formulation will depend on the particular predictive learner $\ell \in \{\mathrm{nw}, \mathrm{nn}\}$ considered in its construction. Should it exist, we will denote the worst-case model of the maximization problem in (15) as $\mathbb{M}^\ell$. When interpreting D as a distance function between models, the worst-case model is that model close to the training model which is however maximally adversarial with respect to the cost of decisions as estimated using a particular predictive learner. This worst-case model may give us additional insight into which training data point are most significant. As we will indicate, the worst-case model can often be computed at no additional cost when solving the maximization problem in (15).

3.1 The Robust Nadaraya-Watson Formulation

We make the previous generic robust supervised learning formulations concrete first in the context of the nominal Nadaraya-Watson formulation. Afterwards we will do the same in the context of the nearest neighbors formulation as well. The resulting robust Nadaraya-Watson formulation can be represented as a tractable convex optimization problem for arbitrary model distance functions. As not to interrupt the discussion, the proof of Theorem 1 is deferred to the appendix.

Theorem 1 (Robust Nadaraya-Watson Formulation)**.** *Consider the abstract robust supervised learning formulation in the context of the Nadaraya-Watson learner $\mathbb{Y}_n^{\mathrm{nw}}$ given in Definition 5. The corresponding robust Nadaraya-Watson formulation as defined in Definition 10 can be reformulated as the convex optimization problem*

$$
\begin{array}{rll}
z_n^{\mathrm{r,nw}}(\bar{x}, \mathbb{M}_{n,t}) := z_{n,t}^{\mathrm{r,nw}}(\bar{x}) \in \arg\min_z \; c_n^{\mathrm{r,nw}}(z, \mathbb{M}_{n,t}, \bar{x}) := \sup & \sum_{\mathtt{M}_n} S_n(x - \bar{x}) \cdot L(z, y) \cdot \mathbb{P}(x, y) & \\
\text{s.t.} & s \in \mathtt{R}_{++}, \;\; \mathbb{P} : \mathtt{M}_n \to \mathtt{R}_+, & \\
& \sum_{\mathtt{M}_n} \mathbb{P}(x, y) = s, & (16) \\
& \sum_{\mathtt{M}_n} S_n(x - \bar{x}) \cdot \mathbb{P}(x, y) = 1, & \\
& s \cdot D(\mathbb{P}/s, \mathbb{M}_{n,t}) \leq s \cdot r_n. &
\end{array}
$$

Proof. Remark that substituting the Nadaraya-Watson predictive learner defined in (11) into the definition of the robust supervised learning formulation given in (15) yields the optimization problem $c_n^{\mathrm{r,nw}}(z, \mathbb{M}_{n,t}, \bar{x}) := \sup_{s>0,\, \mathbb{M} \in \mathcal{M}_n} \left\{ s \cdot \sum_{\mathtt{M}_n} S_n(x - \bar{x}) \cdot L(z, y) \cdot \mathbb{M}(x, y) \;:\; s \cdot \sum_{\mathtt{M}_n} S_n(x - \bar{x}) \cdot \mathbb{M}(x, y) = 1,\; D(\mathbb{M}, \mathbb{M}_{n,t}) \leq r_n \right\}$. This previous optimization formulation over the parameter s and model $\mathbb{M}$ is unfortunately nonconvex. The final convex optimization reformulation (16) is then obtained by the nonlinear change of variables $\mathbb{P} := s \cdot \mathbb{M}$. The resulting optimization problem is indeed convex as all equality constraints are linear. The ultimate constraint is obtained via the chain of equivalences $D(\mathbb{M}, \mathbb{M}_{n,t}) \leq r_n \iff D(\mathbb{P}/s, \mathbb{M}_{n,t}) \leq r_n \iff s \cdot D(\mathbb{P}/s, \mathbb{M}_{n,t}) \leq s \cdot r_n$ for $s > 0$. The model distance function D is convex in its first argument and so is its perspective function $s \cdot D(\mathbb{P}/s, \mathbb{M}_{n,t})$ for $s > 0$. □

In the proof of the previous theorem, we show that the worst-case model $\mathbb{M}^{\mathrm{nw}}$ as defined earlier is easily deduced from an optimal solution of the convex reformulation given in (16). The worst-case model is related to the optimal solution $(s^{\mathrm{nw}}, \mathbb{P}^{\mathrm{nw}})$ of our reformulation as the simple equality

$$\mathbb{P}^{\mathrm{nw}} = s^{\mathrm{nw}} \cdot \mathbb{M}^{\mathrm{nw}}.$$

At optimality the normalization variable s^{nw} is the unique solution to the implicit equation (11b) for the worst-case model $\mathbb{M}^{\mathrm{nw}}$. Hence, for those interested the worst-case Nadaraya-Watson model, $\mathbb{M}^{\mathrm{nw}}$ can be computed at no additional cost after having solved the convex optimization reformulation (16).

The maximization problem in (16) characterizing the robust Nadaraya-Watson formulation is concave. Its first optimization variable s is merely one dimensional, while its second optimization variable $\mathbb{P}$ mapping the support $\mathtt{M}_n$ to $\mathtt{R}_+$ can suitably be represented using a vector of $|\mathtt{M}_n|$ positive numbers. Its ultimate constraint is the only nonlinear one and is convex as the perspective function $s \cdot D(\mathbb{P}/s, \mathbb{M}_{n,t})$ is convex jointly in both variables whenever the model distance function D considered is. The bootstrap robust Nadaraya-Watson learner $c_n^{\mathrm{r,nw}}$ hence evaluates the cost of a fixed decision z using the empirical training model $\mathbb{M}_{n,t}$ in the covariate context $\bar{x}$ by solving a finite dimensional convex optimization problem. Remark again that the smoother weights $S_n(x - \bar{x})$ are always positive. As a result, the robust cost $c_n^{r,\mathrm{nw}}(z, \mathbb{M}_{n,t}, \bar{x})$ is a convex function of the decisions z for any empirical training model and covariate context when the loss function satisfies Assumption 1. Thus, the robust Nadaraya-Watson prescription $z_{n,t}^{\mathrm{r,nw}}(\bar{x})$ is characterized as the minimum to a convex function.

Whether or not the robust Nadaraya-Watson cost and prescriptor are tractable ultimately depends on whether the distance function D and the loss function L are efficiently representable (Ben-Tal, El Ghaoui, and Nemirovski, 2009). Throughout this paper we assume that this is indeed the case.

3.2 The Robust Nearest Neighbors Formulation

We specialize now the generic robust supervised learning formulations to the context of the nearest neighbors learning as well. The robust nearest neighbors formulation can again be represented as a convex optimization problem for any arbitrary model distance function. However, its reformulation will be slightly more involved though still tractable. We have in Section 2 divided the support $\mathtt{M}_n$ of the training data into the nested neighborhoods $\mathtt{N}_n(\bar{x}, j)$. Each of these neighborhoods sets contains those points in the support closest to the context of interest $\bar{x}$. The neighborhood parameter j ranges from one to the number of distinct training data points $|\mathtt{M}_n|$. A particular neighborhood $\mathtt{N}_n(\bar{x}, j)$ then contained the k_n nearest neighbors around $\bar{x}$ of the empirical data, if and only if, the empirical model $\mathbb{M}_{n,t}$ is contained in its corresponding model set $\mathcal{N}_n(\bar{x}, j)$ as defined in (13c). We associate with each of these mutually exclusive model sets the partial robust cost

$$\begin{array}{rll} c_{n,j}^{\mathrm{r,nn}}(z, \mathbb{M}_n, \bar{x}) := \sup & \sum_{\mathtt{N}_n(\bar{x},j)} S_n(x - \bar{x}) \cdot L(z, y) \cdot \mathbb{P}(x, y) & \\ \text{s.t.} & s \in \mathtt{R}_{++}, \ \mathbb{P} : \mathtt{M}_n \to \mathtt{R}_+, & \\ & \sum_{\mathtt{N}_n(\bar{x},j)} S_n(x - \bar{x}) \cdot \mathbb{P}(x, y) = 1, & (17) \\ & \sum_{\mathtt{M}_n} \mathbb{P}(x, y) = s, \ \mathbb{P} \in \mathcal{N}_n(\bar{x}, j) \cdot s, & \\ & s \cdot D(\mathbb{P}/s, \mathbb{M}_n) \leq s \cdot r_n. & \end{array}$$

Determining this partial robust cost for a fixed decision z reduces to solving a convex optimization problem in the same primal variables s and $\mathbb{P}$ as was the case in the robust Nadaraya-Watson formulation. In fact, both optimization problems are equivalent bar the convex penultimate constraint in (17). This penultimate constraint is indeed convex and, following (13c), equivalent to the polyhedral condition

$$\mathbb{P} \in \mathcal{N}_n(\bar{x}, j) \cdot s \iff s \cdot \tfrac{k_n}{n} \leq \textstyle\sum_{\mathtt{N}_n(\bar{x},j)} \mathbb{P}(x, y), \ s \cdot \tfrac{k_n - 1}{n} \geq \textstyle\sum_{\mathtt{N}_n(\bar{x},j-1)} \mathbb{P}(x, y).$$

Again, positivity of the smoother weights $S_n(x - \bar{x})$ guarantees that each of the partial robust costs is a convex function of the decision z for a given empirical model $\mathbb{M}_n$ and covariate context $\bar{x}$. Using these partial robust costs we can obtain a tractable formulation of the robust nearest neighbors formulation.

Theorem 2 (Robust nearest neighbors formulation)**.** *Consider the abstract robust supervised learning formulation in the context of the nearest neighbors contextual learner $\mathbb{Y}_n^{\mathrm{nn}}$ given in Definition 7. The robust nearest neighbors formulation can be reformulated as the convex optimization problem*

$$z_n^{\mathrm{r,nn}}(\bar{x}, \mathbb{M}_{n,t}) := z_{n,t}^{\mathrm{r,nn}}(\bar{x}) \in \min_z c_n^{\mathrm{r,nn}}(z, \mathbb{M}_{n,t}, \bar{x}) := \max_{j \in \{1,\ldots,|\mathtt{M}_n|\}} c_{n,j}^{\mathrm{r,nn}}(z, \mathbb{M}_{n,t}, \bar{x}). \tag{18}$$

Proof. Remark that substituting the nearest neighbors learner defined in (13) into the definition of the robust supervised learning formulation given in (15) yields the optimization problem $c_n^{\mathrm{r,nn}}(z, \mathbb{M}_n, \bar{x}) = \sup_{j \in \{1,\ldots,|\mathtt{M}_n|\}, s>0, \mathbb{M} \in \mathcal{M}_n} \{s \cdot \sum_{\mathtt{N}_n(\bar{x},j)} S_n(x - \bar{x}) \cdot L(z, y) \cdot \mathbb{M}(x, y) : s \cdot \sum_{\mathtt{N}_n(\bar{x},j)} S_n(x - \bar{x}) \cdot \mathbb{M}(x, y) = 1, \ \mathbb{M} \in \mathcal{N}_n(\bar{x}, j), \ D(\mathbb{M}, \mathbb{M}_n) \leq r_n\}$. This previous optimization formulation over the parameters j and s and the model $\mathbb{M}$ is unfortunately nonconvex. In fact, it is an integer optimization problem due to the integrality of the neighborhood parameter j. As before we use the nonlinear change of variables $\mathbb{P} := s \cdot \mathbb{M}$. and arrive at the equivalent optimization formulation $\max_{j \in \{1,\ldots,|\mathtt{M}_n|\}} \sup_{s>0, \mathbb{P}:\mathtt{M}_n \to \mathtt{R}_+} \{\sum_{\mathtt{N}_n(\bar{x},j)} S_n(x - \bar{x}) \cdot L(z, y) \cdot \mathbb{P}(x, y) : \sum_{\mathtt{N}_n(\bar{x},j)} S_n(x - \bar{x}) \cdot \mathbb{P}(x, y) = 1, \sum_{\mathtt{M}_n} \mathbb{P}(x, y) = s, \mathbb{P} \in \mathcal{N}_n(\bar{x}, j) \cdot s, \ s \cdot D(\mathbb{P}/s, \mathbb{M}_n) \leq s \cdot r_n\} = \max_{j \in \{1,\ldots,|\mathtt{M}_n|\}} c_{n,j}^{\mathrm{r,nn}}(z, \mathbb{M}_n, \bar{x})$. □

The final robust nearest neighbors formulation consists hence of the maximum of the $|\mathtt{M}_n|$ partial functions discussed before. Each of these partial costs is itself characterized as a convex maximization problem in the same variables $\mathbb{P}$ and s as was the case in the robust Nadaraya-Watson formulation. As a finite maximum of convex robust partial cost functions, the final robust nearest neighbors formulation is evidently convex in the decision z as well. Hence, determining its associated prescription $z_{n,t}^{\mathrm{r,nn}}(\bar{x})$ in the context of the training data only requires solving a convex optimization problem. The robust nearest neighborhood cost function and prescriptor are tractable to compute due to our assumptions on the distance function D and the loss function L as efficiently representable (Ben-Tal, El Ghaoui, and Nemirovski, 2009).

In the proof of Theorem 2, we show that the worst-case model $\mathbb{M}^{\text{nn}}$ as defined earlier in the context of robust nearest-neighbors learning is again easily deduced from an optimal solution of the reformulation given in Theorem 1. Let us denote with $\mathbb{P}_j^{\text{nn}}$ and s_j^{nn} the associated optimal solutions in the convex formulations (17) associated with the model set $\mathcal{N}_n(\bar{x}, j)$. Furthermore, let j^{nn} be an index such that $c_n^{\text{r,nn}}(z, \mathbb{M}_{n,t}, \bar{x}) = c_{n,j^{\text{nn}}}^{\text{r,nn}}(z, \mathbb{M}_{n,t}, \bar{x})$. The worst-case model $\mathbb{M}^{\text{nn}}$ in (15) is then related as

$$\mathbb{P}_{j^{\text{nn}}}^{\text{nn}} = s_{j^{\text{nn}}}^{\text{nn}} \cdot \mathbb{M}^{\text{nn}}.$$

At optimality, the normalization variable $s_{j^{\text{nn}}}^{\text{nn}}$ and neighborhood parameters j^{nn} are the unique solution to the implicit equations (13b) and (13c) defining the nearest neighbors learner for the worst-case model $\mathbb{M}^{\text{nn}}$. Hence for those interested in the worst-case Nadaraya-Watson model, $\mathbb{M}^{\text{nn}}$ can once more be computed at no additional cost after having solved the convex optimization reformulation (18).

In this section, we were merely interested in the practical viability of the generic robust supervised learning formulations stated in Definition 10. We argued that for arbitrary convex model distance functions this is indeed the case. Unfortunately, most convex model distance functions do not necessarily guarantee that the corresponding generic robust supervised learning formulations perform well on the out-of-sample bootstrap data. In the next section, we will single out one particular distance function for which this is nevertheless the case. Correspondingly, we will come to denote this special model distance function as the bootstrap distance function.

4 Bootstrap Prescriptive Performance

Robustness to overfitting must be understood, not as a property of a particular prescription, but rather as a property of a prescriptor. Indeed, robustness to overfitting is a quality of a function mapping data to prescriptions rather than the quality of a particular prescription itself. In Definition 4, we called a prescriptive method bootstrap robust against overfitting if it makes prescriptions which do well on a large fraction $1-b$ of the bootstrap data $\mathtt{M}_{n,b}$. Good performance on bootstrap data does evidently not guarantee good performance on real out-of-sample data. The relevance of our bootstrap robustness guarantee indeed stands or falls with the extent to which it can generate bootstrap data which resembles actual out-of-sample data. Judging, however, by the tremendous practical success of the bootstrap procedure, we take here its efficacy at face value. In some sense, bootstrap data is the closest we can hope to get to actual test data without making statistical assumption of how the data came to be.

Furthermore, it must be remarked that a small bootstrap disappointment (10) does not mean that the supervised learning formulation performs well in any absolute sense. It merely means the robust formulation does well when compared to its nominal counterpart on data artificially obtained using the bootstrap procedure. It hence goes without saying that if the nominal supervised learning formulation itself is unsatisfactory, adding bootstrap robustness by itself will not suffice. Keeping these precautions in mind we point out that those looking for absolute guarantees will not find them in data alone.

In the previous section we indicated that a learning formulation which is robust with respect to any arbitrary model distance function is not necessarily bootstrap robust in the sense of Definition 4. In the remainder of this section we will indicate that for the following particular model distance function this is however nevertheless the case. We also provide an even more practical representation of the particular robust supervised learning formulations based on convex duality specific to this particular bootstrap distance function.

Definition 11 (The Bootstrap Distance Function)**.** *For two empirical models $\mathbb{M}_n$ and $\mathbb{M}'_n$ in $\mathcal{M}_n$ we define their bootstrap distance as*

$$B(\mathbb{M}_n, \mathbb{M}'_n) := \sum_{\mathtt{M}_n} \mathbb{M}_n(x, y) \cdot \log\left(\frac{\mathbb{M}_n(x, y)}{\mathbb{M}'_n(x, y)}\right). \tag{19}$$

The bootstrap distance between two empirical models is recognized as the relative entropy distance for discrete distributions as stated in Table 1. The relative entropy is also known as information for discrimination, cross-entropy, information gain or Kullback-Leibler divergence (Kullback and Leibler, 1951).

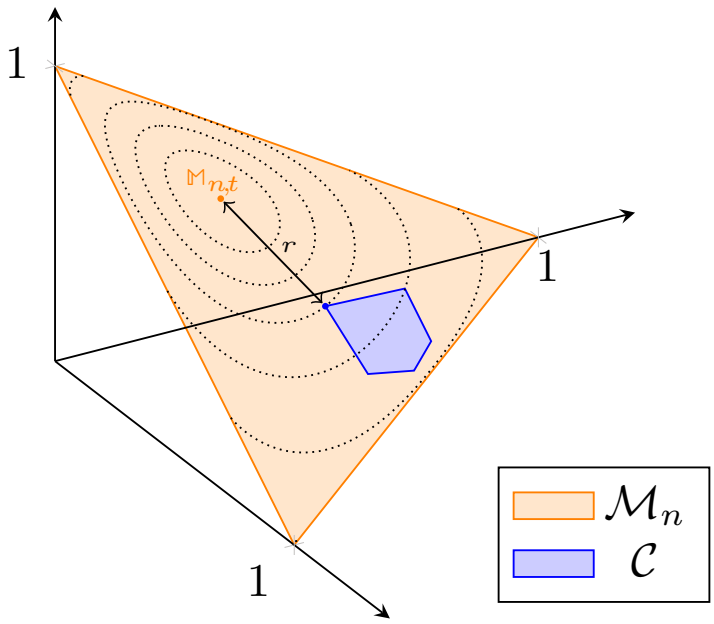


Figure 3: Visualization of the bootstrap inequality (20) in Theorem 3. The probability $\mathbb{M}_{n,t}^n(\mathbb{M}_{n,b} \in \mathcal{C})$ decays at the exponential rate $r := \inf_{\mathbb{M}\in\mathcal{C}} B(\mathbb{M}, \mathbb{M}_{n,t})$, which can be viewed as the bootstrap *distance* of the empirical training model $\mathbb{M}_{n,t}$ to the set of interest $\mathcal{C}$. The triangle visualizes the probability simplex of all empirical models $\mathcal{M}_n$ supported on the support on by the training data.

In order to proof that the bootstrap distance function results in robust supervised learning formulations who suffer a limited bootstrap disappointment we will only need one elementary result from large deviation theory. The following theorem characterizes the essential large deviation behavior of the bootstrap sample distribution $\mathbb{M}_{n,b}$. This result forms the backbone of most of the theoretical results in this paper concerning the statistical properties of our supervised learning formulations.

Theorem 3 (The Bootstrap Inequality (Csiszár, 1984, Theorem 1))**.** *The probability that the random bootstrap model $\mathbb{M}_{n,b}$ of the random bootstrap model $\mathbb{M}_{n,b}$ realizes in a convex set of models $\mathcal{C}$ satisfies the finite sample inequality*

$$\mathbb{M}_{n,t}^n(\mathbb{M}_{n,b} \in \mathcal{C}) \leq \exp\left(-n \cdot \inf_{\mathbb{M}\in\mathcal{C}} B(\mathbb{M}, \mathbb{M}_{n,t})\right), \quad \forall n \geq 0. \tag{20}$$

The geometry of the bootstrap inequality is visualized in Figure 3. The bootstrap inequality is of high-quality and is asymptotically exact in the exponential rate. Large deviation theory concerns itself (Csiszár, 1984) indeed with the corresponding lower bound

$$-\inf_{\mathbb{M}\in\operatorname{int}\mathcal{C}} B(\mathbb{M}, \mathbb{M}_{n,t}) \leq \liminf_{n\to\infty} \frac{1}{n} \log \mathbb{M}_{n,t}^n(\mathbb{M}_{n,b} \in \mathcal{C}) \tag{21}$$

which meets the upper bound (20) asymptotically in its the exponential rate for regular event sets $\mathcal{C} = \operatorname{cl}\operatorname{int}\mathcal{C}$ as the bootstrap distance function is continuous in its first argument because its second happens to be $\mathbb{M}_{n,t}(x, y) > 0$ for any $(x, y) \in \mathbb{M}_{n,t}$.

4.1 A Bootstrap Robust Nadaraya Watson Formulation

A generic robust Nadararya-Watson formulation may not necessarily suffer a small bootstrap disappointment when the model distance function is ill-chosen. We use the large deviation inequality from Theorem 3 to quantify the bootstrap disappointment which the bootstrap robust Nadaraya-Watson formulation suffers when using the bootstrap distance function B given in Definition 11. Afterwards we will establish a similar result for the nearest neighbors formulation as well.

Theorem 4 (Bootstrap Performance of the Nadaraya-Watson Formulation)**.** *The robust Nadaraya-Watson formulation* (16) *with bootstrap distance function* ($D = B$) *suffers bootstrap disappointment* (10) *at most* $b = \exp(-n \cdot r_n)$.

Proof. Let us fix the covariate context $\bar{x}$, training data set $\mathtt{M}_{n,t}$ and decision z. In order to prove the theorem it suffices to characterize the probability of the event that the bootstrap model $\mathbb{M}_{n,b}$ of the random bootstrap data $\mathtt{M}_{n,b}$ realizes in the set of distributions

$$\begin{aligned}\mathcal{C} &= \left\{\mathbb{M} \in \mathcal{M}_n \;:\; c(z, \mathbb{Y}^{\mathrm{nw}}(\bar{x}, \mathbb{M})) > c^{\mathrm{r,nw}}_{n,t}\right\}, \\ &= \left\{\mathbb{M} \in \mathcal{M}_n \;:\; \exists s > 0,\; s \cdot \textstyle\sum_{\mathtt{M}_n} S_n(x - \bar{x}) \cdot L(z, y) \cdot \mathbb{M}(x, y) > c^{\mathrm{r,nw}}_{n,t},\; s \cdot \sum_{\mathtt{M}_n} S_n(x - \bar{x}) \cdot \mathbb{M}(x, y) = 1\right\}.\end{aligned}$$

The first equality follows immediately from the definition of the supervised Nadaraya-Watson learning algorithm given in (11). After eliminating the auxiliary variable s in the last description of the set $\mathcal{C}$, we arrive at

$$\mathcal{C} = \left\{\mathbb{M} \in \mathcal{M}_n \;:\; \textstyle\sum_{\mathtt{M}_n} S_n(x - \bar{x}) \cdot L(z, y) \cdot \mathbb{M}(x, y) > c^{\mathrm{r,nw}}_{n,t} \cdot \sum_{\mathtt{M}_n} S_n(x - \bar{x}) \cdot \mathbb{M}(x, y)\right\}.$$

The last characterization of the set $\mathcal{C}$ also shows that it is convex. In fact, it shows that the event set $\mathcal{C}$ is a polyhedral set. The robust cost $c^{\mathrm{r,nw}}_{n,t}$ is constructed precisely to ensure that for any model distance function D we have $\inf_{\mathbb{M} \in \mathcal{C}} D(\mathbb{M}, \mathbb{M}_{n,t}) > r_n$. Indeed, $\bar{\mathbb{M}} \in \mathcal{C} \iff c(z, \mathbb{Y}^{\mathrm{nw}}(x, \bar{\mathbb{M}})) > c^{r,\mathrm{nw}}_{n,t} := \sup\{c(z, \mathbb{Y}^{\mathrm{nw}}(\bar{x}, \mathbb{M})) \;:\; D(\mathbb{M}, \mathbb{M}_{n,t}) \leq r\} \implies D(\bar{\mathbb{M}}, \mathbb{M}_{n,t}) > r$. Hence, the final result follows from the bootstrap inequality (20) as $\mathcal{C}$ is a convex set and in this particular case the employed model distance function ($D = B$) coincides with the bootstrap distance function (19). □

In order thus to be guaranteed a bootstrap disappointment b, the bootstrap robustness radius should hence be scaled as $r_n = \frac{1}{n} \log \frac{1}{b}$ when an increasing amount of training data gets available. Despite being tractable, the robust Nadaraya-Watson formulation is still stated in (16) as the solution to a saddle point problem which may be awkward to handle. Furthermore, the size of the inner maximization problem characterizing the bootstrap robust Nadaraya-Watson formulation grows with the number of distinct training data samples $|\mathtt{M}_n|$. Thus, finding the robust Nadaraya-Watson prescription $z^{r,\mathrm{nn}}_{n,t}(\bar{x})$ may become a daunting endeavor when the training data set contains a huge amount of distinct samples. The following lemma shows that both of these problems can be somewhat alleviated when working with the bootstrap distance function B as a model distance function.

Lemma 1 (Dual Representation of the Bootstrap Robust Nadaraya-Watson Cost)**.** *The bootstrap robust cost* $c^{\mathrm{r,nw}}_n : \mathtt{Z} \times \mathcal{M}_n \times \mathtt{X} \to \mathtt{R}$ *with respect to the bootstrap distance function* B *can be represented using a dual convex optimization problem as*

$$\begin{array}{ll} \inf & \alpha \\ \text{s.t.} & \alpha \in \mathtt{R},\; \nu \in \mathtt{R}_+, \\ & \nu \cdot \log\left(\sum_{\mathtt{M}_n} \exp\left((L(z, y) - \alpha) \cdot S_n(x - \bar{x})/\nu\right) \cdot \mathbb{M}_n(x, y)\right) + r_n \cdot \nu \leq 0. \end{array} \tag{22}$$

when the robustness radius $r_n > 0$ *is strictly positive.*

The dual characterization of the bootstrap robust Nadaraya-Watson cost function amounts to a convex optimization problem. The dual characterization requires the Slater condition $r_n > 0$ to hold in order to be equivalent to the original primal characterization. We remark that in the one case ($r_n = 0$) the Slater condition does not hold, the robust Nadaraya-Watson formulation collapses to the nominal one due to the discrimination property of the nonparametric bootstrap distance function. The main advantage of using the dual formulation stated in Lemma 1 is that finding the optimal prescription $z^{r,\mathrm{nw}}_{n,t}(\bar{x})$ now merely requires the solution of a convex optimization problem jointly over the decision z and dual variables α and β, instead of a saddle point problem with variables of a dimension which may scale linearly in the amount of training data. The dual formulation requires two one dimensional dual variables independent of the amount of training data. Its dependence on the amount of training data is not completely absent, though, as the constraint in the dual characterization (22) counts $|\mathtt{M}_n|$ terms. All in all, we have shown the robust Nadaraya-Watson formulation to be tractable and suffering only a bounded bootstrap disappointment when using the bootstrap distance function B. We will do now the same for the nearest neighbors formulation as well.

4.2 A Bootstrap Robust Nearest Neighbors Formulation

We again use the large deviation inequality from Theorem 3 to quantify the bootstrap disappointment which the robust nearest neighbors formulation suffers when using the bootstrap distance function B. Notice that for the optimization problem defining the partial cost $c^{\mathrm{r,nn}}_{n,j}$ to be nontrivial on the training model $\mathbb{M}_{n,t}$, the robustness radius r_n needs to be bigger than the minimum robustness radius

$$\begin{array}{lll} r^\star_{j,n} := \inf & D(\mathbb{M}, \mathbb{M}_{n,t}) & \\ \text{s.t.} & \mathbb{M} : \mathtt{M}_n \to \mathtt{R}_+, & \\ & \sum_{\mathtt{M}_n} \mathbb{M}(x,y) = 1,\ \mathbb{M} \in \mathcal{N}_n(\bar{x}, j). & \end{array} \tag{23}$$

If this is the case, then the feasible set of the optimization problem (17) defining the partial cost $c^{\mathrm{r,nn}}_{n,j}$ is indeed non-empty. Notice that also the minimum bootstrap radii are characterized as the solution of tractable convex optimization problem over the convex model sets $\mathcal{N}_n(\bar{x}, j)$. Its optimization variable $\mathbb{M}$ mapping the support $\mathtt{M}_n$ to $\mathtt{R}_+$ can suitably be represented as a vector of $|\mathtt{M}_n|$ positive numbers. These minimum bootstrap radii play an important role in the characterization of the bootstrap disappointment suffered by the nearest neighbors formulation.

Theorem 5 (Bootstrap Performance of the Nearest Neighbors Formulation)**.** *The bootstrap robust nearest neighbors formulation* (18) *with bootstrap distance function ($D = B$) suffers bootstrap disappointment* (10) *at most* $b = \sum_{j \in \{1,\ldots,|\mathtt{M}_n|\}} \exp\left(-n \cdot \max\{r_n, r^\star_{j,n}\}\right)$.

Proof. Let us fix the covariate context $\bar{x}$, training data set $\mathtt{M}_{n,t}$ and decision z. In order to prove the theorem, it suffices to characterize the probability of the event that the empirical model $\mathbb{M}_{n,b}$ of the random bootstrap data $\mathbb{M}_{n,b}$ realizes in the set of models $\mathcal{C} = \left\{\mathbb{M} \in \mathcal{M} \ : \ c(z, \mathbb{Y}^{\mathrm{nn}}(\bar{x}, \mathbb{M})) > c^{\mathrm{r,nn}}_{n,t}\right\} = \cup_{j^\star \in \{1,\ldots,|\mathtt{M}_n|\}} \mathcal{C}_j$ with

$$\mathcal{C}_j := \left\{ \mathbb{M} \in \mathcal{N}_n(\bar{x}, j) \ : \ \begin{array}{l} \exists s > 0,\ s \cdot \sum_{\mathtt{N}_n(\bar{x},j)} S_n(x - \bar{x}) \cdot L(z,y) \cdot \mathbb{M}(x,y) > c^{\mathrm{r,nn}}_{n,t}, \\ s \cdot \sum_{\mathtt{N}_n(\bar{x},j)} S_n(x - \bar{x}) \cdot \mathbb{M}(x,y) = 1 \end{array} \right\}.$$

The first equality follows immediately from the definition of the supervised nearest neighbors learning algorithm given in (13). After eliminating the auxiliary variable s we arrive at the descriptions $\mathcal{C}_j = \{\mathbb{M} \in \mathcal{N}_n(\bar{x}, j) : \sum_{\mathtt{N}_n(\bar{x},j)} S_n(x - \bar{x}) \cdot L(z,y) \cdot \mathbb{M}(x,y) > c^{\mathrm{r,nn}}_{n,t} \cdot \sum_{\mathtt{N}_n(\bar{x},j)} S_n(x - \bar{x}) \cdot \mathbb{M}(x,y)\}$. Each of the partial sets $\mathcal{C}_j$ is a convex polyhedron. We can use the union bound to establish

$$\mathbb{M}^n_{n,t}(\mathbb{M}_{n,b} \in \mathcal{C}) \leq \sum_{j^\star \in \{1,\ldots,|\mathtt{M}_n|\}} \mathbb{M}^n_{n,t}(\mathbb{M}_{n,b} \in \mathcal{C}_j).$$

The partial robust nearest neighbors costs $c^{\mathrm{r,nn}}_{n,j}$ are constructed to ensure that $\inf_{\mathbb{M} \in \mathcal{C}_j} D(\mathbb{M}, \mathbb{M}_{n,t}) > r_n$. By virtue of $\mathcal{C}_j \subseteq \mathcal{N}_n(\bar{x}, j)$, evidently, we must also have that $\inf_{\mathbb{M} \in \mathcal{C}_j} D(\mathbb{M}, \mathbb{M}_{n,t}) > r^\star_{j,n} := \inf\{D(\mathbb{M}, \mathbb{M}_{n,t}) : \mathbb{M} \in \mathcal{N}_n(\bar{x}, j)\}$. Indeed, we have the rather direct implication $\bar{\mathbb{M}} \in \mathcal{C}_j \implies c(z, \mathbb{Y}^{\mathrm{nn}}(x, \bar{\mathbb{M}})) > c^{\mathrm{r,nn}}_{n,t} \geq \sup\{c(z, \mathbb{Y}^{\mathrm{nn}}(\bar{x}, \mathbb{M})) \,|\, D(\mathbb{M}, \mathbb{M}_{n,t}) \leq r,\ \mathbb{M} \in \mathcal{N}_n(\bar{x}, j)\}$ which in turn itself implies $D(\bar{\mathbb{M}}, \mathbb{M}_{n,t}) > r$. Hence, the result follows from the bootstrap inequality (20) applied to each of the probabilities $\mathbb{M}^n_{n,t}(\mathbb{M}_{n,b} \in \mathcal{C}_j)$ and in this particular case the employed model distance function ($D = B$) coincides with the bootstrap distance function (19). □

The previous theorem gives an explicit characterization of the bootstrap performance of the nearest neighbors formulation. Choosing the robustness radius r_n yielding a desired bootstrap disappointment b can not be done analytically, but thanks to the convex characterization (23) of the minimum bootstrap radii $r^\star_{j,n}$ it can nevertheless be carried out numerically in a tractable fashion.

Despite all previous encouraging result regarding the bootstrap performance of the robust nearest neighbors formulation, it is still stated as the solution to a saddle point problem in (17) which may be awkward to handle practically. Here both the size and the number of the maximization problems constituting the bootstrap robust nearest neighbors formulation grows linearly with the amount of distinct training data samples $|\mathtt{M}_n|$. The following lemma tries to alleviate one of these concerns by considering a dual formulation of the maximization problem characterizing the partial robust cost functions.

Lemma 2 (Dual Representation of the Bootstrap Robust Nearest Neighbors Cost)**.** *The partial bootstrap robust cost* $c^{\mathrm{r,nn}}_{n,j} : \mathtt{Z} \times \mathcal{M}_n \times \mathtt{X} \to \mathtt{R}$ *can be represented using a dual convex optimization problem as*

$$
\begin{array}{ll}
\inf & \alpha \\
\text{s.t.} & \alpha \in \mathtt{R},\ \eta \in \mathtt{R}^2_+,\ \nu \in \mathtt{R}_+, \\
& \quad \nu \log \Big(\sum_{\mathtt{N}_n(\bar{x},j-1)} \exp([(L(z,y)-\alpha)\cdot S_n(x-\bar{x}) + \eta_1 - \eta_2]/\nu) \cdot \mathbb{M}_n(x,y) \\
& \qquad + \sum_{\mathtt{N}_n(\bar{x},j)\setminus \mathtt{N}_n(\bar{x},j-1)} \exp([(L(z,y)-\alpha)\cdot S_n(x-\bar{x}) + \eta_1]/\nu) \cdot \mathbb{M}_n(x,y) \\
& \qquad + \sum_{\mathtt{M}_n \setminus \mathtt{N}_n(\bar{x},j)} \mathbb{M}_n(x,y) \Big) + r_n \cdot \nu - \frac{k_n}{n}(\eta_1 - \eta_2) - \frac{\eta_2}{n} \leq 0.
\end{array}
\tag{24}
$$

when the robustness radius satisfies $r_n > r^\star_{j,n}$.

The main advantage of using this convex dual formulation of the robust nearest neighbors formulation is that finding the optimal prescription $z^{r,\mathrm{nn}}_{n,t}(\bar{x})$ now merely requires the solution of a convex optimization problem over the decision z and this time three additional dual variables α, β, and η, instead of a saddle point problem with variables of a dimension which may scale linearly in the amount of training data. This dependence on the amount of training data is again not completely eliminated as the constraint in the dual characterization (24) of the partial robust cost $c^{\mathrm{r,nn}}_{n,j}$ still counts j terms. As the final robust nearest neighbors cost function $c^{\mathrm{r,nn}}_n$ consists of the maximum of these partial robust cost functions we still have to account for a total number of $\frac{1}{2}|\mathtt{M}_n|(|\mathtt{M}_n|+1)$ such terms.

5 Numerical Examples

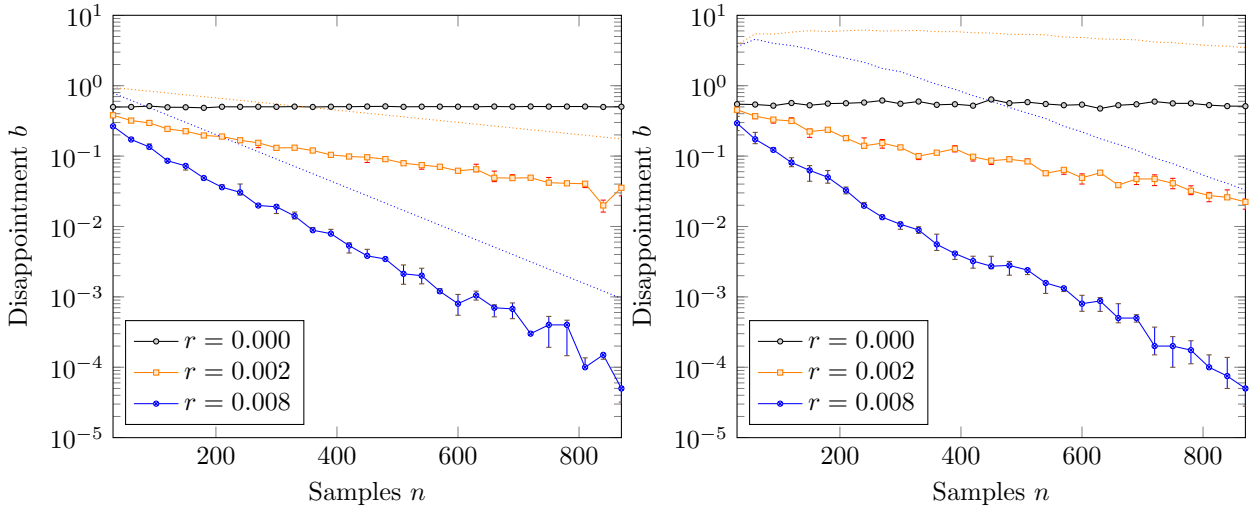
We discuss a data-driven news vendor problem in Section 5.1 and a data-driven portfolio allocation problem in Section 5.2. Both of these problems are prescriptive analytics problems stated generally in (4) for a particular loss function L. For both problems, we consider the nominal and bootstrap robust supervised learning formulations discussed in this paper. We briefly discuss first how our supervised learning formulations were solved and trained in practice. All algorithms were implemented in `Julia` (Bezanson et al., 2017).

The nominal Nadaraya-Watson and nearest neighbors formulations of Bertsimas and Kallus, 2014 were implemented with the help of the `Convex` package developed by Udell et al. (2014). Taking advantage of the dual representations given in Lemmas 1 and 2, the same procedure was followed for their robust counterparts with respect to the bootstrap distance function as well. The corresponding exponential cone optimization problems were solved numerically with the `ECOS` interior point solver by Domahidi, Chu, and Boyd (2013).

Both the Nadaraya-Watson and nearest neighbors formulations require several hyper parameters such as the smoother function S or the number of neighbors to be learned from data. We will use synthetic training data based on a known model $\mathbb{M}^\star$ which allows us to generate as much data as desired. We considered a Nadaraya-Watson formulation using the Gaussian smoother function given in Figure 1. Likewise, we considered the classical nearest neighbors formulation with the Mahalanobis distance metric $d(m=(x,y),\bar{x}) = (x-\bar{x})^\top \Sigma^{-1}_{n,t}(x-\bar{x})$ based on the empirical variance $\Sigma_{n,t} := \sum_{\mathtt{M}_{n,t}} (x-\mu_{n,t})\cdot(x-\mu_{n,t})^\top/n$ and the empirical mean of the auxiliary data $\mu_{n,t} := \sum_{\mathtt{M}_{n,t}} x/n$. Potential ties among equidistant points were broken based on the size of their labels. The bandwidth parameter h_n and the number of nearest neighbors k_n were determined based on the squared prediction loss performance of the corresponding Nadaraya-Watson or nearest neighbors predictive learner on ten data sets cross validated from the training data.

5.1 A news vendor problem

A company sells a perishable good and needs to make an order $z \in \mathtt{R}$. Ideally, the company would of course like to order exactly $z = Y$ where Y is the demand of the perishable good. Unfortunately, a decision on the order quantity needs to be made before the demand is observed. Fortunately, however, the company can observe before making the order several covariates $X = \bar{x}$ which may correlate with the uncertain demand.



(a) Nadaraya-Watson Formulations

(b) Nearest Neighbors Formulations

Figure 4: The empirical bootstrap disappointment b of the Nadaraya-Watson and nearest neighbors formulations in function of the number of samples n. The nominal Nadaraya-Watson and nearest neighbors formulation corresponds to the case $r = 0$. Such nominal formulations do not safeguard against over-calibration as they disappoint on random bootstrap data about half ($b \approx \frac{1}{2}$) the time. The dotted lines visualize the upper bounds concerning the bootstrap disappointment of the bootstrap robust Nadaraya-Watson and nearest neighbors formulation given in Theorem 4 and Theorem 5, respectively. Large deviation theory (Csiszár, 1984) indicates that these bootstrap upper bounds and the actual bootstrap disappointments of either formulation drop to zero at the same exponential rate r.

The company may consider the day of the week $D \in \{\text{Monday}, \ldots, \text{Sunday}\}$ to capture weekly cyclical demand, and the outside temperature $T \in \mathtt{R}$ which can influence demand as well. Here, only two covariates are considered where in practice many more may be taken into consideration. For repetitive sales, a sensible goal is to order a quantity that minimizes the total expected cost according to

$$z^\star(\bar{x}) \in \arg\inf \quad \mathrm{E}_{\mathbb{M}^\star}\left[L(z, Y) := b \cdot (Y - z)^+ + h \cdot (z - Y)^+ \middle| X = \bar{x}\right].$$

The constants $b = 10 \in \mathtt{R}_+$ and $h = 1 \in \mathtt{R}_+$ represent here the marginal cost in American dollar of back ordering and holding goods. If the model distribution $\mathbb{M}^\star$ is known, then a classical result states that the optimal decision is then given by the quantile $z^\star(\bar{x}) := \inf\left\{z \,:\, \mathrm{E}_{\mathbb{M}^\star}\left[\mathbb{1}\{Y \leq z\} \middle| X = \bar{x}\right] \geq b/b + h\right\}$ of the demand distribution in the covariate context of interest. The classical news vendor formulation assumes the joint distribution $\mathbb{M}^\star$ between returns and covariates to be known. In practice however this is almost never the case.

A supervised data version of this news vendor problem is discussed by Rudin and Vahn (2014) in which instead historical data $\mathtt{M}_{n,t}$ is given consisting of historical demands $\mathtt{Y}_{n,t}$ and covariates $\mathtt{X}_{n,t}$. In this data-driven setting, we resort to the supervised data formulations presented by Bertsimas and Kallus, 2014 and their robust counterparts derived in this work. It should be remarked that the data-driven formulation of Rudin and Vahn (2014) is not at all similar to ours, but instead is based on an empirical risk minimization approach. We consider synthetic training data drawn as independent samples from the synthetic model $\mathbb{M}^\star$ with a Gaussian conditional distribution

$$\mathbb{Y}^\star(\bar{x} = (\bar{t}, \bar{d})) = N\big(100 + (\bar{t} - 20) + 20 \cdot \mathbb{I}(\bar{d} \in \{\text{Weekend}\}), 16\big)$$

and where the day of the week and outside temperature are independent random variables distributed uniformly and normally as $N(20, 4)$, respectively. We shall use this synthetic big data news vendor problem to illustrate the bootstrap disappointment of the robust Nadaraya-Watson and nearest neighbors formulations in a particular context of interest, e.g., $\bar{x} = (\bar{t}, \bar{d}) = (15°\text{C}, \text{Friday})$.

We would like to investigate to what extent our bootstrap robust formulations prevent against overfitting the training data set $\mathtt{M}_{n,t}$. Given a budgeted cost $c^{r,\ell}_{n,t}$ and action $z^{r,\ell}_{n,t}$ calibrated to this training data set, we approximate its bootstrap disappointment as stated in Definition 4 using a large number $m = 20,000$ of bootstrap resamples. In Figure 4, we present this empirical bootstrap disappointment as a function of the number n of training samples for the nominal and robust Nadaraya-Watson and nearest neighbors formulations. The nominal Nadaraya-Watson and nearest neighbors formulation corresponds to the case $r = 0$. Such nominal formulations do not safeguard against over-calibration as they disappoint on random bootstrap data about half the time. The dotted lines visualize the upper bounds concerning the bootstrap disappointment of the bootstrap robust Nadaraya-Watson and nearest neighbors formulation given in Theorem 4 and Theorem 5, respectively. The guarantee in case of the nearest neighbors formulation is not as tight as its Nadaraya-Watson counterpart mostly due to the use of the union bound in the proof of Theorem 5. Nevertheless, large deviation theory via (21) ensures that the empirical bootstrap disappointments and their corresponding theoretical upper bound in either formulation drop to zero at the same exponential rate r.

5.2 A portfolio allocation problem

We consider a portfolio allocation problem in which the decision $z \in \mathtt{R}^6_+$ consists in how to split a limited investment budget among each of six securities in an artificial portfolio. The returns $Y \in \mathtt{R}^6$ that each of those securities will provide is evidently uncertain and not known ahead of time. These uncertain returns may furthermore be indirectly affected by a large number of covariates X. Investment returns may be influenced by the global $\mathtt{S\&P500} \in \mathtt{R}$ performance and other general market indicators such as the inflation $I \in \mathtt{R}$. When the artificial portfolio contains any defense contractor securities, we might also want to include the amount of Twitter chatter mentioning the hash tag $\mathtt{\#WAR} \in \mathtt{R}_+$ as a crude geopolitical indicator. All three covariates may have an indirect impact on the returns of each of the securities in our portfolio. Evidently, before any investment is made it would be wise to take the current market performance and geopolitical situation $X = \bar{x}$ into account.

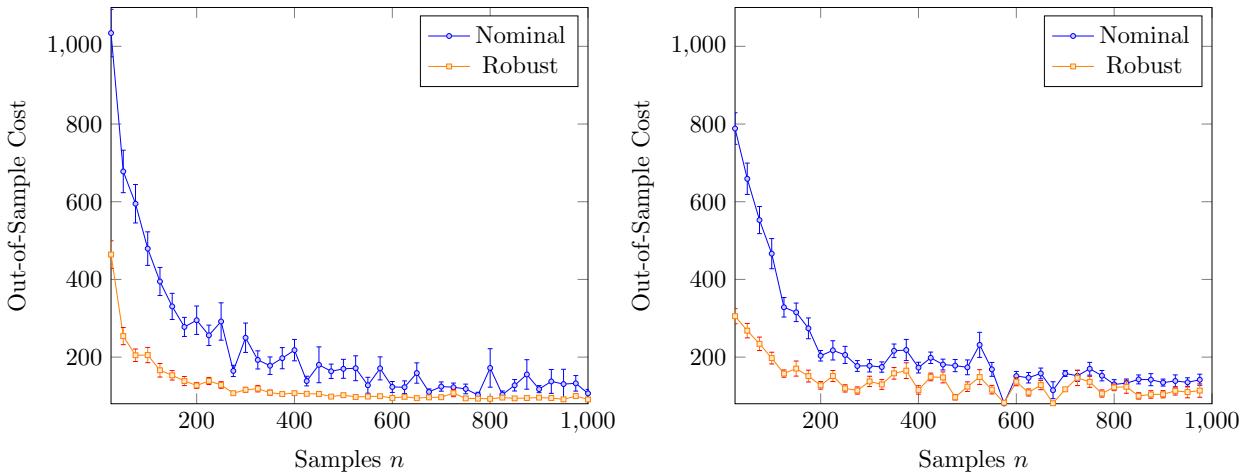
The classical formulation of such portfolio allocation problems hypothesizes that the returns Y and covariates X to be random variables distributed jointly according to a statistical model $\mathbb{M}^\star$. The investor seeks to maximize the mean return $\mathrm{E}_{\mathbb{M}^\star}\left[z^\top Y | X = \bar{x}\right]$ while minimizing the risk that the loss $(-z^\top Y)^+ := \max\{-z^\top Y, 0\}$ is exceedingly large in the covariate context $\bar{x}$. Following a reformulation of conditional value-at-risk due to Rockafellar and Uryasev (2000), we can consider the conditional value-at-risk of negative returns at risk level ϵ using an auxiliary decision variable β as the convex minimization problem $\inf_\beta \beta + \frac{1}{\epsilon}\mathrm{E}_{\mathbb{M}^\star}\left[(-z^\top Y - \beta)^+ | X = \bar{x}\right]$. The last risk measure comes with the intuitive interpretation as the expected tail loss occurring above the $(1-\epsilon)$ quantile. Using a trade off $\lambda \in \mathtt{R}_+$ between risk and return our final formulation of the portfolio allocation problem reads

$$(z^\star(\bar{x}), \beta^\star(\bar{x})) \in \arg\inf_{(z \in \mathtt{R}_+,\, \beta \in \mathtt{R})} \left\{ \mathrm{E}_{\mathbb{M}^\star}\left[L(z, \beta, Y) = \beta + \tfrac{1}{\epsilon}(-z^\top Y - \beta)^+ - \lambda \cdot z^\top Y | X = \bar{x}\right] : \mathbb{1}^\top z = 1 \right\}. \quad (25)$$

The larger the trade off parameter the less important the risk of incurring losses becomes in favor of the expected return. In the extreme case $\lambda = 0$, the investor would only invest in the least risky security while for $\lambda \to \infty$ only the security promising the maximal mean return would be considered. By varying the value of the trade off parameter any preferred risk return trade off can be investigated. Here, we consider their particular values $\epsilon = 0.05$ and $\lambda = 1$ exclusively.

The classical portfolio allocation formulation (25) assumes the exact statistical model between returns and covariates to be known. In practice however this is seldom the case. Instead of a statistical model, merely historical data $\mathtt{M}_{n,t}$ consisting of historical observations $\mathtt{Y}_{n,t}$ and covariates $\mathtt{X}_{n,t}$ can reasonably be assumed to be given in practice. We will consider synthetic training data drawn as independent samples from a synthetic model $\mathbb{M}^\star$ with the Gaussian conditional distribution

$$\mathbb{Y}^\star(\bar{x} = (\overline{\mathtt{sap500}}, \bar{i}, \overline{\mathtt{\#war}})) = N\big(\mu + 0.1 \cdot (\overline{\mathtt{sap500}} - 1000) \cdot \mathbb{1}_6 + 1000 \cdot \bar{i} \cdot \mathbb{1}_6 + 10 \cdot \log(\overline{\mathtt{\#war}} + 1) \cdot \mathbb{1}_6,\, \Sigma\big)$$



(a) Nadaraya-Watson Formulations

(b) Nearest Neighbors Formulations

Figure 5: Out-of-sample cost of the robust Nadaraya-Watson and nearest neighbors prescriptions $z_{n,t}^{r,\ell}$ as a function of amount n of training data. As more data becomes available either supervised learning formulation improves and asymptotically converges to the ground truth given in (25). Robust prescriptions nevertheless significantly outperform their nominal counterparts when the amount of data is limited.

and with the nominal mean μ and the covariance matrix Σ as

$$\mu = \begin{pmatrix} 86.8625 & 71.6059 & 75.3759 & 97.6258 & 52.7854 & 84.8973 \end{pmatrix}^\top,$$

$$\Sigma^{1/2} = \begin{pmatrix} 136.687 & \star & \star & \star & \star & \star \\ 8.79766 & 142.279 & \star & \star & \star & \star \\ 16.1504 & 15.0637 & 122.613 & \star & \star & \star \\ 18.4944 & 15.6961 & 26.344 & 139.148 & \star & \star \\ 3.41394 & 16.5922 & 14.8795 & 13.9914 & 151.732 & \star \\ 24.8156 & 18.7292 & 17.1574 & 6.36536 & 24.7703 & 144.672 \end{pmatrix}.$$

The covariates `SAP500`, I and $\log(\texttt{\#WAR})$ are all independent random variables distributed as $N(1000, 50)$, $N(0.02, 0.01)$ and $N(0, 1)$, respectively. We shall use this synthetic big data portfolio problem to illustrate the actual out-of-sample costs of the robust Nadaraya-Watson and nearest neighbors formulations in a particular context of interest, e.g., $\bar{x} = (\overline{\texttt{sap500}}, \bar{i}, \overline{\texttt{\#war}}) = (970, 0, 10)$.

In Figure 5, we depict the out-of-sample cost of the nominal prescription $z_{n,t}^{r,\ell}$ according to both the nominal Nadaraya-Watson and nearest neighbors formulation ($r = 0$) as a function of the size of the training data set. This out-of-sample cost was computed as the average cost of the nominal prescription on a thousand data sets containing synthetically generated test data. The curve itself represents the average of fifty random training data sets to make sure the reported results are statistically significant rather than a luck of the draw. As one would expect, the actual out-of-sample cost of the prescribed actions reduces as the formulations have access to more data. As all supervised learning formulations discussed in this paper are statistically consistent their out-of-sample costs converge eventually to the same asymptotic cost.

The typical behavior which illustrates the value of considering prescriptions which perform well not only on the training data but on bootstrap data as well can be seen in the same figure too. On the same figure we illustrate the out-of-sample performance of the robust decisions $z_{n,t}^{r,\ell}$ with respect to the bootstrap distance function with that robustness radius r guaranteeing a bootstrap disappointment of at most $b = 0.01$. It should be remarked that robust prescriptions always have a higher cost on the training data than their nominal counterparts. They are indeed less calibrated to this particular data set than their nominal counterparts. On the other hand, their robustness enables them to enjoy a better out-of-sample performance. As expected the benefits of robustness diminish as more training data becomes available. Robustness to overfitting is indeed most useful when the training data set is limited.

6 Conclusion

We discussed in this paper prescriptive analytics problems where cost optimal decisions are to be adapted to a specific covariate context using only supervised data. Supervised learning formulations allow for superior context specific decision-making when compared to the naive sample average formulation. As all data-driven methods are prone to adverse overfitting phenomena we must safeguard against over-calibration to one particular training data set. To that end we introduced a novel notion of robustness which guards against overfitting and crucially is itself completely data-driven. Our notion of bootstrap robustness is inspired by the statistical bootstrap, and does not pose any statistical assumption on training data. We derived bootstrap robust learning formulations which are as tractable as their nominal counterparts based on ideas from distributionally robust optimization. Finally, we have illustrated the benefits of bootstrap robust decisions empirically in terms of their superior out-of-sample performance on a small news vendor problem as well as a small portfolio allocation problem.

Acknowledgments

The second author is generously supported by the Early Post.Mobility fellowship No. 165226 of the Swiss National Science Foundation.

References

Altman, N.S. (1992). "An introduction to kernel and nearest-neighbor nonparametric regression". In: *The American Statistician* 46.3, pp. 175–185.

Ben-Tal, A., L. El Ghaoui, and A. Nemirovski (2009). *Robust Optimization*. Princeton University Press.

Bertsimas, D., V. Gupta, and N. Kallus (2014). "Robust SAA". In: *ArXiv*. URL: https://arxiv.org/abs/1408.4445.

Bertsimas, D. and N. Kallus (2014). "From predictive to prescriptive analytics". In: *ArXiv*. URL: https://arxiv.org/abs/1402.5481.

Bezanson, J., A. Edelman, S. Karpinski, and V.B. Shah (2017). "Julia: A fresh approach to numerical computing". In: *SIAM Review* 59.1, pp. 65–98.

Csiszár, I. (1984). "Sanov property, generalized I-projection and a conditional limit theorem". In: *The Annals of Probability* 12.3, pp. 768–793.

Delage, E. and Y. Ye (2010). "Distributionally robust optimization under moment uncertainty with application to data-driven problems". In: *Operations Research* 58.3, pp. 595–612.

Domahidi, A., E. Chu, and S. Boyd (2013). "ECOS: An SOCP solver for embedded systems". In: *European Control Conference (ECC)*, pp. 3071–3076.

Efron, B. (1982). *The Jackknife, the Bootstrap and Other Resampling Plans*. SIAM.

Epanechnikov, V.A. (1969). "Non-parametric estimation of a multivariate probability density". In: *Theory of Probability & Its Applications* 14.1, pp. 153–158.

Erdoğan, E. and G. Iyengar (2006). "Ambiguous chance constrained problems and robust optimization". In: *Mathematical Programming* 107.1–2, pp. 37–61.

Friedman, J., T. Hastie, and R. Tibshirani (2001). *The Elements of Statistical Learning*. Vol. 1. Springer Series in Statistics. Springer.

Kullback, S. and R. A. Leibler (1951). "On information and sufficiency". In: *The Annals of Mathematical Statistics* 22.1, pp. 79–86.

Li, Q. and J.S. Racine (2007). *Nonparametric Econometrics: Theory and Practice*. Princeton University Press.

Michaud, R. O. (1989). "The Markowitz optimization enigma: Is 'optimized' optimal?" In: *Financial Analysts Journal* 45.1, pp. 31–42.

Mohajerin Esfahani, P. and D. Kuhn (2015). "Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations". In: *ArXiv*. URL: https://arxiv.org/abs/1505.05116.

Nadaraya, E.A. (1964). "On estimating regression". In: *Theory of Probability & Its Applications* 9.1, pp. 141–142.

Nemirovski, A. and A. Shapiro (2006). "Convex approximations of chance constrained programs". In: *SIAM Journal on Optimization* 17.4, pp. 969–996.

Pflug, G. and D. Wozabal (2007). "Ambiguity in portfolio selection". In: *Quantitative Finance* 7.4, pp. 435–442.

Postek, K., D. den Hertog, and B. Melenberg (2016). "Computationally tractable counterparts of distributionally robust constraints on risk measures". In: *SIAM Review* 58.4, pp. 603–650.

Rockafellar, R.T. and S. Uryasev (2000). "Optimization of conditional value-at-risk". In: *Journal of risk* 2, pp. 21–42.

Rudin, C. and G.-Y. Vahn (2014). "The big data newsvendor: Practical insights from machine learning". In: *SSRN*. URL: https://dx.doi.org/10.2139/ssrn.2559116.

Scott, D.W. (2015). *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons.

Shapiro, A. (2003). "Monte Carlo sampling methods". In: *Handbooks in Operations Research and Management Science* 10, pp. 353–425.

Shapiro, A., D. Dentcheva, and A. Ruszczyńsk (2014). *Lectures on Stochastic Programming: Modeling and Theory*. SIAM.

Silverman, B.W. (1986). *Density Estimation for Statistics and Data Analysis*. Vol. 26. CRC press.

Stellato, B., B.P.G. Van Parys, and P.J. Goulart (2017). "Multivariate Chebyshev inequality with estimated mean and variance". In: *The American Statistician* 71.2, pp. 123–127.

Sun, H. and H. Xu (2016). "Convergence analysis for distributionally robust optimization and equilibrium problems". In: *Mathematics of Operations Research* 41.2, pp. 377–401.

Udell, M., K. Mohan, D. Zeng, J. Hong, S. Diamond, and S. Boyd (2014). "Convex optimization in Julia". In: *SC14 Workshop on High Performance Technical Computing in Dynamic Languages*.

Van Parys, B.P.G., P.M. Esfahani, and D. Kuhn (2017). "From data to decisions: Distributionally robust optimization is optimal". In: *ArXiv*. URL: https://arxiv.org/abs/1505.05116.

Van Parys, B.P.G., P.J. Goulart, and D. Kuhn (2016). "Generalized Gauss inequalities via semidefinite programming". In: *Mathematical Programming* 156.1-2, pp. 271–302.

Van Parys, B.P.G., P.J. Goulart, and M. Morari (2015). "Distributionally robust expectation inequalities for structured distributions". In: *Optimization Online*.

Wang, Z., P.W. Glynn, and Y. Ye (2016). "Likelihood robust optimization for data-driven newsvendor problems". In: *Computational Management Science* 12.2, pp. 241–261.

Watson, G.S. (1964). "Smooth regression analysis". In: *Sankhyā: The Indian Journal of Statistics, Series A* 26.4, pp. 359–372.

A Proofs

A.1 Proof of Lemma 1

Proof. We will employ standard Lagrangian duality on the convex optimization characterization (16) of the Nadaraya-Watson cost function. The Lagrangian function associated with the primal optimization problem in (16) is denoted here at the function

$$\mathcal{L}(\mathbb{P}, s; \alpha, \beta, \nu) := \textstyle\sum_{\mathtt{M}_n} S_n(x - \bar{x}) \cdot L(z, y) \cdot \mathbb{P}(x, y) + \left(1 - \sum_{\mathtt{M}_n} S_n(x - \bar{x}) \cdot \mathbb{P}(x, y)\right) \alpha + \\ \left(\sum_{\mathtt{M}_n} \mathbb{P}(x, y) - s\right) \beta + \left(r_n \cdot s - \sum_{\mathtt{M}_n} \mathbb{P}(x, y) \log\left(\tfrac{\mathbb{P}(x,y)}{s \cdot \mathbb{M}_n(x,y)}\right)\right) \nu$$

where $\mathbb{P}$ and s are the primal variables of the primal optimization problem (16) and α, β and ν the dual variables associated with each of its constraints. Collecting the relevant terms in the Lagrangian function results in

$$\mathcal{L}(\mathbb{P}, s; \alpha, \beta, \nu) = \alpha + s(r_n \nu - \beta) + \textstyle\sum_{\mathtt{M}_n} \left[\mathbb{P}(x, y) \left((L(z, y) - \alpha) \cdot S_n(x - \bar{x}) + \beta\right) - \nu \mathbb{P}(x, y) \log\left(\tfrac{\mathbb{P}(x,y)}{s \cdot \mathbb{M}_n(x,y)}\right)\right]$$

The dual function of the primal optimization problem (16) is identified with the concave function $g(\alpha, \beta, \nu) := \inf_{\mathbb{P} \geq 0,\, s > 0} \mathcal{L}(\mathbb{P}, s; \alpha, \beta, \nu)$. Our dual function can be expressed alternatively as $g(\alpha, \beta, \nu) =$

$$\sup_{s>0} \alpha + s\left(r_n \nu - \beta\right) + \sup_{\mathbb{P} \geq 0} \textstyle\sum_{\mathtt{M}_n} \left[\mathbb{P}(x, y) \left((L(z, y) - \alpha) \cdot S_n(x - \bar{x}) + \beta\right) - \nu \mathbb{P}(x, y) \log\left(\tfrac{\mathbb{P}(x,y)}{s \cdot \mathbb{M}_n(x,y)}\right)\right]$$

$$= \sup_{s>0} \alpha + s\left(r_n \nu - \beta\right) + \textstyle\sum_{\mathtt{M}_n} \sup_{\mathbb{P}(x,y) \geq 0} \left[\mathbb{P}(x, y) \left((L(z, y) - \alpha) \cdot S_n(x - \bar{x}) + \beta\right) - \nu \mathbb{P}(x, y) \log\left(\tfrac{\mathbb{P}(x,y)}{s \cdot \mathbb{M}_n(x,y)}\right)\right]$$

$$= \sup_{s>0} \alpha + s\left(r_n \nu - \beta\right) + s \textstyle\sum_{\mathtt{M}_n} \mathbb{M}_n(x, y) [\sup_{\lambda \geq 0} \lambda \left((L(z, y) - \alpha) \cdot S_n(x - \bar{x}) + \beta\right) - \nu \lambda \log(\lambda)].$$

The inner maximization problems over λ can be dealt with using the Fenchel conjugate of the $\lambda \mapsto \lambda \cdot \log \lambda$ function as

$$= \sup_{s>0} \alpha + s\left(r_n \nu - \beta\right) + s\nu \textstyle\sum_{\mathtt{M}_n} \mathbb{M}_n(x, y) \exp\left(\frac{(L(z,y) - \alpha) \cdot S_n(x - \bar{x}) + \beta}{\nu} - 1\right)$$

$$= \left\{\alpha \;:\; r\nu + \nu \textstyle\sum_{\mathtt{M}_n} \mathbb{M}_n(x, y) \exp\left(\frac{(L(z,y) - \alpha) \cdot S_n(x - \bar{x}) + \beta}{\nu} - 1\right) \leq \beta\right\}.$$

The dual optimization problem of the primal problem (16) is now found as $\inf_{\alpha,\beta,\nu \geq 0} g(\alpha, \beta, \nu)$. As the primal optimization in (16) is convex, strong duality holds under Slater's condition which is satisfied whenever $r > 0$. Using first-order optimality conditions, the optimal $\beta^\star$ must satisfy the relationship $\beta^\star = -\nu + \nu \log\left(\sum_{\mathtt{M}_n} \mathbb{M}_n(x, y) \exp((L(z, y) - \alpha) \cdot S_n(x - \bar{x})/\nu)\right)$. Substituting the optimal value of $\beta^\star$ in the back in the dual optimization problem gives

$$\inf_{\alpha,\beta,\nu \geq 0} g(\alpha, \beta, \nu) = \inf_{\alpha,\nu \geq 0} g(\alpha, \beta^\star, \nu)$$

$$= \inf\left\{\alpha \in \mathtt{R} \;:\; \exists \nu \in \mathtt{R}_+,\, r\nu + \nu \log\left(\textstyle\sum_{\mathtt{M}_n} \mathbb{M}_n(x, y) \exp\left(\frac{(L(z,y) - \alpha) \cdot S_n(x - \bar{x})}{\nu}\right)\right) \leq 0\right\}.$$

□

A.2 Proof of Lemma 2

Proof. We will employ standard Lagrangian duality on the convex optimization characterization (17) of the partial nearest neighbors cost function associated with the model set $\mathcal{N}_n(\bar{x}, j)$. The Lagrangian function associated with the primal optimization problem in (17) is denoted here at the function

$$\begin{aligned}\mathcal{L}(\mathbb{P}, s; \alpha, \beta, \eta, \nu) := &\textstyle\sum_{\mathtt{N}_n(\bar{x},j)} S_n(x-\bar{x}) \cdot L(z,y) \cdot \mathbb{P}(x,y) + \Big(1 - \sum_{\mathtt{N}_n(\bar{x},j)} S_n(x-\bar{x}) \cdot \mathbb{P}(x,y)\Big)\,\alpha \\ &+ \Big(\textstyle\sum_{\mathtt{N}_n(\bar{x},j)} \mathbb{P}(x,y) - \frac{k_n}{n} \cdot s\Big)\,\eta_1 + \Big(\frac{k_n - 1}{n} \cdot s - \sum_{\mathtt{N}_n(\bar{x},j-1)} \mathbb{P}(x,y)\Big)\,\eta_2 \\ &+ \Big(\textstyle\sum_{\mathtt{M}_n} \mathbb{P}(x,y) - s\Big)\,\beta + \Big(r_n \cdot s - \sum_{\mathtt{M}_n} \mathbb{P}(x,y) \log\Big(\frac{\mathbb{P}(x,y)}{s \cdot \mathbb{M}_n(x,y)}\Big)\Big)\,\nu\end{aligned}$$

where $\mathbb{P}$ and s are the primal variables of the primal optimization problem (16) and α, β, η and ν the dual variables associated with each of its constraints. Collecting the relevant terms in the Lagrangian function results in $\mathcal{L}(\mathbb{P}, s; \alpha, \beta, \nu) =$

$$\begin{aligned}&\alpha + s(r_n\nu - \beta - \tfrac{k_n}{n}(\eta_1 - \eta_2) - \tfrac{\eta_2}{n}) \\ &\quad + \textstyle\sum_{\mathtt{N}_n(\bar{x},j-1)} \Big[\mathbb{P}(x,y)\left((L(z,y) - \alpha) \cdot S_n(x-\bar{x}) + \beta + \eta_1 - \eta_2\right) - \nu \mathbb{P}(x,y) \log\Big(\frac{\mathbb{P}(x,y)}{s \cdot \mathbb{M}_n(x,y)}\Big)\Big] \\ &\quad + \textstyle\sum_{\mathtt{N}_n(\bar{x},j) \setminus \mathtt{N}_n(\bar{x},j-1)} \Big[\mathbb{P}(x,y)\left((L(z,y) - \alpha) \cdot S_n(x-\bar{x}) + \beta + \eta_1\right) - \nu \mathbb{P}(x,y) \log\Big(\frac{\mathbb{P}(x,y)}{s \cdot \mathbb{M}_n(x,y)}\Big)\Big] \\ &\quad + \textstyle\sum_{\mathtt{M}_n \setminus \mathtt{N}_n(\bar{x},j)} \Big[\mathbb{P}(x,y)\beta - \nu \mathbb{P}(x,y) \log\Big(\frac{\mathbb{P}(x,y)}{s \cdot \mathbb{M}_n(x,y)}\Big)\Big]\end{aligned}$$

The dual function of the primal optimization problem (16) is identified with the concave function $g(\alpha, \beta, \eta, \nu) := \inf_{\mathbb{P} \geq 0,\, s > 0} \mathcal{L}(\mathbb{P}, s; \alpha, \beta, \nu)$. Using the same manipulations as presented in the proof of Lemma 1 we can express the dual function as $g(\alpha, \beta, \eta, \nu) =$

$$\begin{aligned}= \sup_{s>0}\; &\alpha + s(r_n\nu - \beta - \tfrac{k_n}{n}(\eta_1 - \eta_2) - \tfrac{\eta_2}{n}) + s\nu \textstyle\sum_{\mathtt{M}_n \setminus \mathtt{N}_n(\bar{x},j)} \mathbb{M}_n(x,y) \exp\Big(\frac{\beta}{\nu} - 1\Big) \\ &+ s\nu \textstyle\sum_{\mathtt{N}_n(\bar{x},j-1)} \mathbb{M}_n(x,y) \exp\Big(\frac{(L(z,y)-\alpha) \cdot S_n(x-\bar{x}) + \beta + \eta_1 - \eta_2}{\nu} - 1\Big) \\ &+ s\nu \textstyle\sum_{\mathtt{N}_n(\bar{x},j) \setminus \mathtt{N}_n(\bar{x},j-1)} \mathbb{M}_n(x,y) \exp\Big(\frac{(L(z,y)-\alpha) \cdot S_n(x-\bar{x}) + \beta + \eta_1}{\nu} - 1\Big).\end{aligned}$$

Our dual function can be expressed alternatively as

$$\begin{aligned}g(\alpha, \beta, \eta, \nu) = \Big\{\alpha : &\, r_n \cdot \nu - \tfrac{k_n}{n}(\eta_1 - \eta_2) - \tfrac{\eta_2}{n} + \nu \textstyle\sum_{\mathtt{M}_n \setminus \mathtt{N}_n(\bar{x},j)} \mathbb{M}_n(x,y) \exp\Big(\frac{\beta}{\nu} - 1\Big) \\ &+ \nu \textstyle\sum_{\mathtt{N}_n(\bar{x},j-1)} \mathbb{M}_n(x,y) \exp\Big(\frac{(L(z,y)-\alpha) \cdot S_n(x-\bar{x}) + \beta + \eta_1 - \eta_2}{\nu} - 1\Big) \\ &+ \nu \textstyle\sum_{\mathtt{N}_n(\bar{x},j) \setminus \mathtt{N}_n(\bar{x},j-1)} \mathbb{M}_n(x,y) \exp\Big(\frac{(L(z,y)-\alpha) \cdot S_n(x-\bar{x}) + \beta + \eta_1}{\nu} - 1\Big) \leq \beta\Big\}.\end{aligned}$$

The dual optimization problem of the primal problem (16) is now found as $\inf_{\alpha,\beta,\nu \geq 0}\; g(\alpha, \beta, \nu)$. As the primal optimization problem in (16) is convex, strong duality holds under Slater's condition which is satisfied whenever $r > r^\star_{j,n}$. Using first-order optimality conditions, the optimal $\beta^\star$ must satisfy the relationship $\beta^\star = -\nu + \nu \log(\sum_{\mathtt{N}_n(\bar{x},j-1)} \mathbb{M}_n(x,y) \exp([(L(z,y)-\alpha) \cdot S_n(x-\bar{x}) + \eta_1 - \eta_2]/\nu) + \sum_{\mathtt{N}_n(\bar{x},j) \setminus \mathtt{N}_n(\bar{x},j-1)} \mathbb{M}_n(x,y) \exp([(L(z,y) - \alpha) \cdot S_n(x-\bar{x}) + \eta_1]/\nu) + \sum_{\mathtt{M}_n \setminus \mathtt{N}_n(\bar{x},j)} \mathbb{M}_n(x,y))$. Substituting the optimal value of $\beta^\star$ in the back in the dual optimization problem gives

$$\begin{aligned}\inf_{\alpha,\beta,\nu \geq 0}\; g(\alpha, \beta, \eta, \nu) = &\inf_{\alpha,\nu \geq 0}\; g(\alpha, \beta^\star, \eta, \nu) \\ = \inf \Big\{\alpha \in \mathtt{R} : &\exists \nu \in \mathtt{R}_+, \exists \eta \in \mathtt{R}_+^2,\; r_n \cdot \nu - \tfrac{k_n}{n}(\eta_1 - \eta_2) - \tfrac{\eta_2}{n} \cdot \nu \\ &+ \nu \log(\textstyle\sum_{\mathtt{N}_n(\bar{x},j-1)} \exp([(L(z,y) - \alpha) \cdot S_n(x-\bar{x}) + \eta_1 - \eta_2]/\nu) \cdot \mathbb{M}_n(x,y) \\ &+ \textstyle\sum_{\mathtt{N}_n(\bar{x},j) \setminus \mathtt{N}_n(\bar{x},j-1)} \exp([(L(z,y) - \alpha) \cdot S_n(x-\bar{x}) + \eta_1]/\nu) \cdot \mathbb{M}_n(x,y) \\ &+ \textstyle\sum_{\mathtt{M}_n \setminus \mathtt{N}_n(\bar{x},j)} \mathbb{M}_n(x,y)) \leq 0\Big\}.\end{aligned}$$

□