Distribution-free Algorithms for Learning Enabled Optimization with Non-parametric Estimation

Shuotao Diao and Suvrajeet Sen *

sdiao@usc.edu, s.sen@usc.edu

March, 2020

Abstract

This paper studies a fusion of concepts from stochastic optimization and non-parametric statistical learning, in which data is available in the form of covariates interpreted as predictors and responses. Such models are designed to impart greater agility, allowing decisions under uncertainty to adapt to the knowledge of the predictors (leading indicators). Specialized algorithms can be looked upon as learning enabled optimization (LEO) algorithms. This paper focuses on equipping LEO with non-parametric estimation approaches (LEON) which provide asymptotically optimal decisions without requiring the specification of a distribution. In particular, our framework accommodates several non-parametric estimation schemes, including k nearest neighbors (kNN), and other standard kernel estimators under one unified framework. Several techniques to improve the quality of decisions are discussed. Finally, we demonstrate the computational performance of Robust LEON-kNN and Robust LEON-kernel for a well-known instance arising in logistics.

Keywords: mini-batch first-order method, stochastic quasi-gradient method, non-parametric statistical estimation, k-NN estimation, kernel estimation

1 Introduction.

Bertsimas and Kallus [2] proposed a new model to accommodate covariates (Z, W) as follows

$$\min_{x \in X} \; f(x, z) = \mathbb{E}[F(x, W)|Z = z]. \tag{1}$$

Here $F(x, \cdot)$ is a real-valued random function defined over the sample space of $W|Z = z$. Unlike ordinary stochastic programming (SP), this model allows data in the form of tuples $\{(z_i, w_i)\}_{i=1}^{\mathcal{N}}$, associated

*Daniel J. Epstein Department of Industrial and Systems Engineering, University of Southern California, Los Angeles CA, 90089-0193, USA

with covariates (Z, W) (i.e., Z is the predictor and W is the response). As shown in Bertsimas and Kallus [2], ordinary Sample Average Approximation (SAA [18, 26]) can produce non-optimal decisions when the conditional distribution of W given $Z = z$ is unknown. They also show that non-parametric estimation can overcome this handicap. On the other hand, without considering predictor-response structure within covariates, the use of non-parametric estimation appears in Hanasusanto and Kuhn [16] and Pflug and Pichler [23]. The former uses kernel regression to estimate the cost-to-go function in approximate dynamic programming, and the latter uses kernel density estimation to approximate the stochastic process in multistage stochastic programming. Depending on the specific structure of the interplay between data and decisions, there are other variations of (1). For readers interested in a general review of solving stochastic optimization problems using machine learning techniques, we recommend [25].

Bertsimas and Kallus [2] proposed a way to approximate (1) by using a weight function as shown below,

$$\min_{x \in X} \ \hat{f}_{\mathcal{N}}(x, z) = \sum_{i=1}^{\mathcal{N}} v_{\mathcal{N},i}(z) F(x, W_i), \tag{2}$$

where $\mathcal{N}$ is the size of dataset $\{(Z_i, W_i)\}_{i=1}^{\mathcal{N}}$. Equation (2) should be considered as a non-parametric analog of SAA and they use supervised machine learning methods, such as k nearest neighbors (kNN), kernel estimation, and random forest, to calculate z-dependent weights $v_{\mathcal{N},i}(z)$. They show that under certain assumptions (e.g., equi-continuity of the random objective function), solving (2) provides a statistically consistent estimate of an optimal solution to (1). In addition, Bertsimas and Kallus [2] propose an index (coefficient of prescriptiveness) which is based on two quantities: a) regret due to imperfect information, b) regret due to unconditional sample average approximation (ignoring dependencies between Z and W). Here the term regret refers to the difference of the objective function estimate of a proposed decision and that obtained by exploiting perfect information. Overall, the objective of [2] is to incorporate non-parametric estimation within a new class of SP models (with covariates in the form of predictors and responses). In this formulation, the power of non-parametric tools helps exploit the information of underlying predictors, which are usually ignored by classical stochastic programming.

In order to motivate this paper, we note that (2) can become very demanding (or even intractable) without appropriate algorithmic support. Non-parametric models generally require large amount of data (large $\mathcal{N}$) and this in turn gives rise to a large-scale SP models which becomes increasingly demanding as $\mathcal{N}$ increases. Further details regarding complexity of calculating kNN version of weight function are provided in Section 4. In an event, there is a need for algorithmic schemes that produce the estimated solutions in a numerically realizable way.

This paper provides a mini-batch first-order method based on non-parametric subgradient estimation

to solve (1). The benefits of using such a method are: a) It avoids depending on the knowledge of conditional probability distributions as required within SP; b) Each estimated solution update is not particularly demanding. Towards this end, the mathematical structure focuses on the weighted average of subgradients where the weights cover a batch (of size N) of data points, instead of the entire data set (i.e. $N << \mathcal{N}$). It is also worth noting that if $v_{N,i}(z) = \frac{1}{N}$ for all $i = 1, 2, ..., N$, then the method reduces to mini-batch stochastic approximation (SA), which is independent of the realization of Z. However, we illustrate via examples that such methods may not converge. In fact, Figure 1 below shows that both mini-batch SA and mini-batch Robust SA fail to converge to the true optimal value/solution of a two-stage shipment problem introduced by [2]. The details associated with this instance are provided below.

Example 1. *[2] Suppose that predictor Z, is a 3-dimensional ARMA(2,2) time series, whose formulation is given below:*

$$Z_t - \phi_1 Z_{t-1} - \phi_2 Z_{t-2} = U_t + \Phi_1 U_{t-1} + \Phi_{t-2} U_{t-2}, \tag{3}$$

where U is a 3-dimensional normal random variable with mean 0 and variance-covariance matrix Σ_U. The response W is a 12 dimensional random vector which is a function of Z and some white noise. The i^{th} component of W, W_i, is written as follows,

$$W_i = \max\{0, A_i^T(Z + \delta_i/4) + (B_i^T Z)\epsilon_i\} \quad i \in \{1, 2, ..., 12\}, \tag{4}$$

where δ_i and ϵ_i are normal random variates with mean 0 and standard deviation 1, and A_i and B_i are given constant vectors. Let $\hat{x}$ be a candidate solution and let $\{W_1^z, ..., W_{\mathcal{N}_v}^z\}$ be the samples generated from the conditional distribution given $Z = z$. *Note that for a synthetic instance such as this, it is possible to choose a validation data set of size $\mathcal{N}_v$ such that the confidence interval of the estimated cost is small enough. In practice, we choose $\mathcal{N}_v$ (which is the size of validation data set) so that the width of confidence interval of the estimated cost is small enough. In particular, for a vector W, its i^{th} component is given below*

$$W_i^z = \max\{0, A_i^T(z + \delta_i/4) + (B_i^T z)\epsilon_i\} \quad i \in \{1, 2, ..., 12\}. \tag{5}$$

The estimated cost of $\hat{x}$ is

$$\theta_v(\hat{x}) = \frac{1}{\mathcal{N}_v} \sum_{i=1}^{\mathcal{N}_v} F(\hat{x}, W_i^z), \tag{6}$$

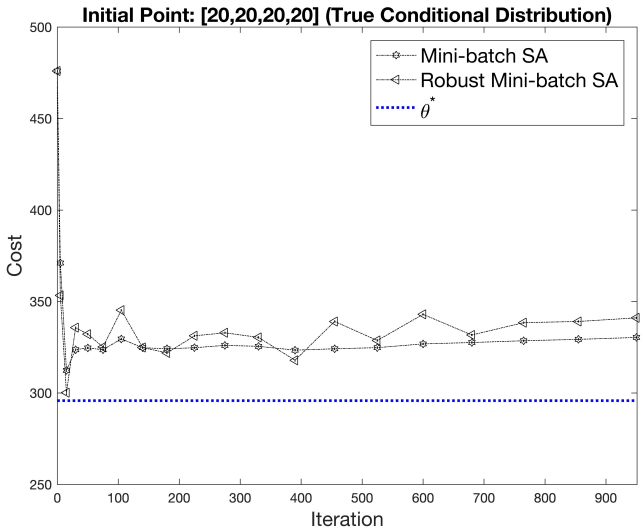
and

$$\theta^* = \min_{x \in X} \frac{1}{\mathcal{N}_v} \sum_{i=1}^{\mathcal{N}_v} F(x, W_i^z) \tag{7}$$

is the estimated "optimal" cost of the perfect-foresight problem. Because the validation data is gener-

ated by assuming the known conditional distribution, the value of θ^* defined in (7) must agree with that obtained by the non-parametric scheme of [2]. The mini-batch SA procedure in this computational experiment is summarized in the Appendix. The estimates shown in Figure 1 reveal that using finite sample

Figure 1: Computational performance of Mini-batch SA and Robust Mini-batch SA in two stage shipment problem



SA or Robust SA is a misguided attempt to estimate the optimal value θ^*. There are several reasons for this failure: a) Problem (1) is not expressed as a finite sum problem. b) It does not exploit the covariate structure of (1).

In view of the above example, one question we wish to answer is: What kind of algorithmic scheme would provide a consistent and asymptotically convergent method? Another question is that can we build an SP algorithm which allows simultaneous estimation and optimization so that the synergy between statistical modeling and optimization algorithms can be exploited in manner that results in a distribution-free modeling-cum-algorithmic process? We will provide a concrete approach to overcome the dilemma posed by non-convergence illustrated in Figure 1. Other than using the non-parametric estimation of the true objective function as in [2], we study a variety of non-parametric methods to estimate the true subgradient for updating the estimated solution in our algorithm, which we refer to as learning enabled optimization with non-parametric estimation (LEON).

In formulating the combined estimation-optimization process, we adopt a k nearest-neighbor (kNN)/ kernel approach for true subgradient estimation and a stochastic quasi-gradient method for optimization ([10]). Despite the fact that kNN/kernel estimates are biased, we will show the convergence of numerical algorithms which yield asymptotic optimality of problems such as (1). It is worth noting that since the response is assumed to be parametrized by the predictor, the convergence of the proposed algorithm

is consistent for all possible realizations of the predictor. Inspired by Polyak's averaging approach in stochastic approximation [22, 24], we utilize the proposed LEON methodology to develop a first-order method, which we refer to as Robust LEON. We will also revisit the computational example provided on the previous page, and show how the Robust LEON algorithm discovers an optimal solution and the optimal value.

The organization of this paper is as follows. In section 2, we present the mathematical setting for LEON and the update rule for the estimated solution based on a combination of first-order method and non-parametric statistical method. In section 3, we provide a Robust LEON algorithm to solve (1). The asymptotic convergence of Robust LEON algorithm is shown at the end of section 3. With the help of the generalized structure of Robust LEON, we also investigate Robust LEON-kNN as well as Robust LEON-kernel in sections 4 and 5, respectively. The computational results of Robust LEON are finally demonstrated in section 6 using the two-stage shipment problem of [2]. These computations justify the labels of Robust LEON applied to methodologies presented in this paper. Overall this paper overcomes an important criticism that input distributions are necessary to even get started with SP models.

2 Mathematical Setting for LEON

In this section, we mainly focus on building the mathematical foundations of LEON, which will be shown to be related to stochastic quasi-gradient method (SQG). Throughout this paper, we use the Euclidean distance to measure the distance between the two predictors. We let $||x||$ denote the Euclidean norm of vector x, and let Π_X denote projection operator onto set X. Let $Z_{[i]}(z)$ denote the i^{th} closest neighbor of z. We use $\mathbb{P}$ and $\mathbb{E}$ denote the probability and expectation operators, respectively.

As outlined earlier, we assume that W depends on Z, and both are continuous random variables which take values in measurable spaces $(\mathbb{R}^{n_z}, \mathcal{R}^{n_z})$ and $(\mathbb{R}^{n_w}, \mathcal{R}^{n_w})$ respectively. Let μ_Z and μ_W denote the marginal distribution functions of Z and W, respectively. Let $p(z, w)$ be the joint density function of Z and W, and assume that the joint distribution π of (Z, W) exists and has the following form,

$$\pi(dz, dw) = p(z, w)dzdw \quad z \in \mathbb{R}^{n_z}, w \in \mathbb{R}^{n_w}$$

Let $m(z) = \int_{\mathbb{R}^{n_w}} p(z, w)dw$ and suppose that the conditional distribution given that $Z = z$ is as follows ([5]).

$$k(z, w) = \begin{cases} \frac{\pi(dz,dw)}{m(z)} & \text{if } m(z) > 0 \\ \int_{\mathbb{R}^{n_z}} p(z', w)\mu(dz') & \text{if } m(z) = 0 \end{cases} \tag{8}$$

Given z as an observation of Z, we study (1) under the following assumptions.

(A0) (Non-negativity of the weight function) Let $v_{N,i} : \mathbb{R}^{d_z} \mapsto \mathbb{R}$. The weight function, $v_{N,i}(z)$, is non-negative for all i.

(A1) (Probabilistic existence of $Z = z$) Although we do not require the knowledge of the distribution underlying the uncertainty, we do need some assumptions about the structure of the distribution. We assume that the joint density function $p(z, w)$ between W and Z exists and the conditional distribution is defined by (8), which obeys $\mathbb{P}(Z \in B_\epsilon(z)) > 0 \quad \forall \epsilon > 0$, where $B_\epsilon(z)$ is a closed ball with radius ϵ centered around z. Note that $p(z, w)$ does not depend on the decision x and hence for the class of problems we study, the distribution underlying the uncertainty is not decision-dependent.

(A2) (i.i.d assumption on data pairs) $(Z_1, W_1), ..., (Z_N, W_N)$ are independent and identically distributed.

(A3) (Compact and convex feasible region) $X \subset \mathbb{R}^n$ is a compact and convex set.

(A4) (Convex objective function and related boundedness properties) Let $F : X \times \mathbb{R}^{n_w} \mapsto \mathbb{R}$. For every $w \in \mathbb{R}^{n_w}$, $F(\cdot, w)$ is convex on X, and its subdifferential is bounded on X almost surely (i.e. there exits $M_G > 0$ such that $||G(x, W)|| \leq M_G < \infty$ a.s.). We also assume that $\mathbb{E}[|F(x, W)|] < \infty$ for all $x \in X$.

(A5) (Lipschitz-type continuity of true objective function) Let $f : X \times \mathbb{R}^{n_z} \mapsto \mathbb{R}$. There exists a continuous function, $C(x) < \infty$, and $p > 0$ such that $|f(x, z_1) - f(x, z_2)| \leq C(x)||z_1 - z_2||^p$, for every $x \in X$ and μ_Z almost all $z_1,\ z_2 \in \mathbb{R}^{n_z}$.

The interplay between the different assumptions reflects the interplay between statistics and optimization in this paper. Assumptions **A0** - **A2** are standard in the statistical learning literature. However, the fact that the data process is independent of the decisions (**A1**) is also common in the SP literature. For decision-dependent uncertainty in the context of simultaneous estimation and optimization, we refer to [20]. Moreover, convexity and boundedness assumptions in **A3** are common in the optimization literature, and similarly assumption **A4** refers to the convexity of a *family of functions* parametrized by w and the subgradients parameterized by w. Thus these assumptions are somewhat different from convexity requirements in standard stochastic programming. As for assumption **A5**, there are several ways to bound the errors from the combination of optimization and non-parametric estimation. Since our algorithm is restricted to first-order approximations, we use Lipschitz-type continuity assumption (**A5**) for z given fixed x in the objective function. By assumption **A5**, for any $x \in X$, $f(x, z)$ is Hölder continuous in z. To compare this assumption with that of Bertsimas and Kallus [2], we note that they allow more general approximations and thus their requirements are less demanding. Focusing further on **A5**, note that while our requirement is stated in a point-wise manner (given x), it is a slight generalization of standard Lipschitz continuity assumptions imposed on various non-parametric methods in

[15]. **A5** is important when analyzing the property of bias in each particular case of non-parametric estimation. Because this brand of Lipschitz continuity is somewhat different from standard Lipschitzian property used in optimization, we will provide an illustrative example in the Appendix.

We begin by stating a result from [8], which defines the comparison between the values of two conditional expectations.

Lemma 1. *[8] Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. Let $\mathcal{F}_0 \subset \mathcal{F}$ and $Y_1, Y_2 \in \mathcal{F}$. Suppose that $\mathbb{E}[|Y_1|] < \infty$ and $\mathbb{E}[|Y_2|] < \infty$. Then $Y_1 \leq Y_2$ implies $\mathbb{E}[Y_1|\mathcal{F}_0] \leq \mathbb{E}[Y_2|\mathcal{F}_0]$ a.s.*

Next, Lemma 2 shows the true objective function and its non-parametric estimation, which are provided in (1) and (2), are convex under the assumptions given above.

Lemma 2 (Convexity of objective functions)**.** *Suppose assumptions **A0** - **A4** are satisfied. Then the following hold:*
(1) $f(\cdot, z)$ is convex on X, for μ_Z almost every $z \in \mathbb{R}^{n_z}$.
(2) $\hat{f}_N(\cdot, z)$ is convex on X, for μ_Z almost every $z \in \mathbb{R}^{n_z}$.

Proof. See the Appendix. ■

Lemmas 1 and 2 establish the structure of the model (convexity) and on a conceptual level, it is amenable to large scale optimization. Moreover, the convexity of both $f(x, z)$ and $\hat{f}_N(x, z)$ with respect to x implies the continuity of both $f(x, z)$ and $\hat{f}_N(x, z)$ with respect to x over the relative interior of X. Another consequence of convexity is that for points where the value is finite, the standard subdifferential of convex analysis exists.

Stochastic quasi-gradient is a generalization of a Monte Carlo sample of a subgradient. Before we provide the specific form of statistical estimate of the subgradient of $f(x, z)$, we give a formal definition of stochastic quasi-gradients below.

Definition 1. *[9] Let $\hat{f}_N(x, z)$ and $\hat{G}_N(x, z)$ be the statistical estimate of $f(x, z)$ and its subgradient, $g(x, z)$, respectively. Furthermore, let x^* be an optimal solution. Then $\hat{G}_N(x, z)$ is the stochastic quasi-gradient of $f(x, z)$ of $f(x, z)$ if it satisfies the following condition:*

$$f(x^*, z) - f(x, z) \geq \langle \mathbb{E}[\hat{G}_N(x, z)], x^* - x \rangle + \tau_N, \quad \text{where } \tau_N \to 0 \text{ as } N \to \infty \tag{9}$$

Unlike Monte Carlo estimates of subgradients, estimation using stochastic quasi-gradient is biased. It is worth noting that inequality (9) is the guiding force for asymptotic convergence of Robust LEON. In the following, we let $G(x, W)$ denote a subgradient of $F(x, W)$. By the scaling and addition properties

of subdifferentials, a subgradient of the non-parametric estimator, $\hat{G}_N(x,z)$, is written as the weighted sum below.

$$\hat{G}_N(x,z) = \sum_{i=1}^{N} v_{N,i}(z) G(x, W_i) \tag{10}$$

Accordingly, a non-parametric estimator (kNN estimator and kernel estimators) is often biased [1]. But under mild assumptions, we show that the bias of both kNN estimator and kernel estimators of true subgradients decrease to zero as the sample size increases to infinity.

Lemma 3. *Suppose assumptions **A0** - **A4** are satisfied. For a given z, we further suppose that $|\mathbb{E}[\hat{f}_N(x,z)] - f(x,z)| \leq \delta_N(x,z)$ for any $x \in X$ and $\delta_N(\cdot,z)$ converges uniformly to 0 on X, as $N \to \infty$. Then $\hat{G}_N(x,z)$ is the stochastic quasi-gradient by definition 1.*

Proof. See the Appendix. ■

It is worth noting that if the assumption of uniform convergence of $\delta_N(x,z)$ is replaced by point-wise convergence, $\hat{G}_N(x,z)$ is still stochastic quasi-gradient of $f(x,z)$. However, uniform convergence of the non-parametric estimation [15] is essential in proving asymptotic convergence of Robust LEON.

We end this section by introducing the update rule for the first-order method equipped with stochastic quasi-gradient.

LEON Update Rule

Inputs: l, x_l, $\gamma_l > 0$, $S^l := \{(z_{l,1}, w_{l,1}), (z_{l,2}, w_{l,2}), ..., (z_{l,N_l}, w_{l,N_l})\}$ (Assume that an oracle generates a dataset, S^l, with N_l pairs of predictors and responses).

1. (Stochastic quasi-gradient calculation) Calculate stochastic quasi-gradient, $\hat{G}_{N_l}(x_l, z)$ by using the formula below:
$$\hat{G}_{N_l}(x_l, z) = \sum_{i=1}^{N_l} v_{N_l,i}(z) G(x_l, W_{l,i}).$$
2. (Estimate solution update) Compute the new estimated solution by using the following formula: $x_{l+1} = \Pi_X(x_l - \gamma_l \hat{G}_{N_l}(x_l, z))$, where $\Pi_X(\cdot)$ denotes the projection operator. $N_{l+1} \leftarrow N_l + \Delta$.

3 Algorithmic Setting for LEON

It turns out that convergence properties of the above update (Basic LEON) rule are not entirely satisfactory because of the biased non-parametric estimates. Consequently, this section presents a "robustified"

algorithm for solving (1). Here, we use the expected optimality gap as the metric to measure the convergence of Robust LEON algorithm, which is formally stated in Theorem 1.

Robust LEON

1. (Initialization) Set the maximum of the norm of the subgradient, M_G. Start with $x_0 \in X$ and $\gamma_0 > 0$ and set $D_X = \max_{x \in X} ||x - x_0||$. Let n and m be some positive integer constants (e.g. $n = 1$, $m = 2$). $q \leftarrow 0$. If $n > 1$, use the **LEON Update Rule** to generate x_1, x_2,...,x_{n-1}. Set $i_q \leftarrow n$ and $j_q \leftarrow n - 1$ (the updates in step 2 will make sure that $i_q < j_q$). Choose the maximum number of outer iterations, q_{max}.
2. (Constant stepsize setup) Set $q \leftarrow q + 1$ and $\gamma \leftarrow \frac{D_X}{M_G\sqrt{mq}}$.
3. (Averaging window setup for inner loops) Set $i_q \leftarrow i_{q-1} + m(q-1)$ and $j_q \leftarrow j_{q-1} + mq$. Start with $x_{i_q - 1}$, use **LEON Update Rule** with fixed step size γ to generate x_{i_q}, x_{i_q+1},..., x_{j_q}.
4. (Averaging) Set $\tilde{x}_q \leftarrow \frac{\sum_{l=i_q}^{j_q} x_l}{mq}$.
5. (Stopping rule) If $q \geq q_{max}$, stop and output the estimated solution. Otherwise, repeat from Step 2.

The motivation for referring to the above update as Robust LEON derives from the step size rule of Robust SA. However, as illustrated in the previous section, Robust SA does not provide a convergent algorithm for (1). By increasing the size of the averaging window (i.e. increase m in mq), we generate a sequence of averaged estimated solutions $\tilde{x}_q$, which is shown to produce an optimality gap which converges to 0.

Before we prove the asymptotic convergence of Robust LEON, we need to show the sufficient reduction by using LEON Update Rule.

Lemma 4 (Sufficient reduction in LEON Update Rule)**.** *Let x_l and x_{l+1} be the l^{th} and $(l+1)^{th}$ iterates, respectively, generated by LEON Update Rule. Suppose that **A0** - **A4** are satisfied. Suppose that there exists $\delta_N(\cdot)$ such that $|\mathbb{E}[\hat{f}_N(x,z)] - f(x,z)| \leq \delta_N(x,z)$ for all $x \in X$. For any $x' \in X$ the following inequality holds:*

$$\begin{aligned}\mathbb{E}[||x_{l+1} - x'||^2] \leq & \mathbb{E}[||x_l - x'||^2] - 2\gamma_l \mathbb{E}[f(x_l, z) - f(x', z)] + \gamma_l^2 M_G^2 \\ & + 2\gamma_l \mathbb{E}[\delta_{N_l}(x_l, z) + \delta_{N_l}(x', z)]\end{aligned}$$

Proof. Since $F(\cdot, W_i)$ is convex on X and $v_{N_l,i} \geq 0$, by Lemma 2, $\hat{f}_N(x,z)$ is convex, which implies that

the following inequality holds,

$$\hat{f}_N(x', z) \geq \hat{f}_N(x_l, z) + (x' - x_l)^\top \hat{G}_{N_l}(x_l, z) \tag{11}$$

By the non-expansiveness property of projection operator, we have $||\Pi_X(x') - \Pi_X(x)|| \leq ||x' - x||$. On the other hand, if $x \in X$, then $\Pi_X(x) = x$. Hence, we have

$$\begin{aligned} ||x_{l+1} - x'||^2 &= ||\Pi_X(x_l - \gamma_l \hat{G}_{N_l}(x_l, z)) - \Pi_X(x')||^2 \\ &\leq ||x_l - \gamma_l \hat{G}_{N_l}(x_l, z) - x'||^2 \\ &= ||x_l - x'||^2 + \gamma_l^2 ||\hat{G}_{N_l}(x_l, z)||^2 - 2\gamma_l (x_l - x')^\top \hat{G}_{N_l}(x_l, z) \end{aligned}$$

Note that there exits $M_G > 0$ such that $||G(x, W)|| \leq M_G < \infty$ a.s.. Therefore, by taking the expectation of the both sides of inequality above, we have

$$\begin{aligned} \mathbb{E}[||x_{l+1} - x'||^2] \leq \mathbb{E}[||x_l - x'||^2] + \gamma_l^2 \mathbb{E}[||\hat{G}_{N_l}(x_l, z)||] \\ - 2\gamma_l \mathbb{E}[(x_l - x')^T \hat{G}_{N_l}(x_l, z)] \qquad (12a) \\ \leq \mathbb{E}[||x_l - x'||^2] + \gamma_l^2 M_G^2 - 2\gamma_l \mathbb{E}[(x_l - x')^\top \hat{G}_{N_l}(x_l, z)] \qquad (12b) \end{aligned}$$

Let S^j be the dataset generated at the j^{th} iteration by using the LEON Update Rule. By taking conditional expectation of (11) on the event that $\{S^j\}_{j=0}^{l-1}$ are given and moving $\hat{f}_N(x_l, z)$ to the left hand side of (11), we get

$$\mathbb{E}[\hat{f}_{N_l}(x', z) - \hat{f}_{N_l}(x_l, z) | S^0, ..., S^{l-1}] \geq \mathbb{E}[(x' - x_l)^\top \hat{G}_{N_l}(x_l, z) | S^0, ..., S^{l-1}] \tag{13}$$

Based on the assumption that $|\mathbb{E}[\hat{f}_N(x, z)] - f(x, z)| \leq \delta_N(x, z)$, we have

$$\begin{aligned} \mathbb{E}[f(x', z) - f(x_l, z) + \delta_{N_l}(x', z) + \delta_{N_l}(x_l, z) | S^0, ..., S^{l-1}] \\ \geq \mathbb{E}[\hat{f}_{N_l}(x', z) - \hat{f}_{N_l}(x_l, z) | S^0, ..., S^{l-1}] \end{aligned} \tag{14}$$

By combining (13) and (14), we obtain

$$\begin{aligned} \mathbb{E}[f(x', z) - f(x_l, z) + \delta_{N_l}(x', z) + \delta_{N_l}(x_l, z) | S^0, ..., S^{l-1}] \\ \geq \mathbb{E}[(x' - x_l)^\top \hat{G}_{N_l}(x_l, z) | S^0, ..., S^{l-1}] \end{aligned} \tag{15}$$

We take the expectation of both sides of the inequality above and obtain

$$\begin{aligned}&\mathbb{E}[f(x',z)-f(x_l,z)+\delta_{N_l}(x',z)+\delta_{N_l}(x_l,z)]\\&\geq \mathbb{E}[(x'-x_l)^\top \hat{G}_{N_l}(x_l,z)]\end{aligned}\tag{16}$$

By using (16) to replace $\mathbb{E}[(x_l-x)^\top G_N(x_l,z)]$ in (12b) with $\mathbb{E}[f(x',z)-f(x_l,z)+\delta_{N_l}(x',z)+\delta_{N_l}(x_l,z)]$, we have

$$\begin{aligned}\mathbb{E}[||x_{l+1}-x'||^2] &\leq \mathbb{E}[||x_l-x'||^2]+\gamma_l^2 M_G^2\\&\quad+2\gamma_l\mathbb{E}[f(x',z)-f(x_l,z)+\delta_{N_l}(x',z)+\delta_{N_l}(x_l,z)]\\&=\mathbb{E}[||x_l-x'||^2]-2\gamma_l\mathbb{E}[f(x_l,z)-f(x',z)]\\&\quad+\gamma_l^2 M_G^2+2\gamma_l\mathbb{E}[\delta_{N_l}(x_l,z)+\delta_{N_l}(x',z)]\end{aligned}$$

This completes the proof. ■

It is worth noting that there are two types of estimated solution updates in Robust LEON algorithm. We refer to the process of using LEON Update Rule and x_l to update x_{l+1} as an iteration of LEON update, while we refer to the process of calculating $\hat{x}_q$ (i.e. running steps 2-5 in Robust LEON) as an iteration of Robust LEON algorithm. It is interesting to note that while neither Robust SA nor Basic LEON provides convergence guarantee for solving (1), the Robust LEON approach does. Notwithstanding the bias associated with stochastic quasi-gradients, Robust LEON assures asymptotic convergence by extending the averaging process in which we use an ever-increasing size of the window. In the following, we use $c^*(z)=f(x^*,z)$.

Theorem 1 (Asymptotic convergence of Robust LEON)**.** *Let z be fixed. Let $\tilde{x}_q$ be generated by Robust LEON, where q stands for the iteration number. Suppose assumptions **A0** - **A5** are satisfied. Suppose that there exists $\delta_N(\cdot)$ such that $|\mathbb{E}[\hat{f}_N(x,z)]-f(x,z)|\leq\delta_N(x,z)$ for all $x\in X$. Further assume that $\delta_N(\cdot,z)\geq\delta_{N+1}(\cdot,z)$ for all $N>0$ and for almost every z, $\delta_N(\cdot,z)$ converges to 0 uniformly on X as N goes to infinity. Let l denote the l^{th} iteration of LEON Update. In each iteration of LEON update (process of generating x_l), a new dataset with larger size is generated. Then the following holds:*

$$\mathbb{E}[f(\tilde{x}_q,z)-c^*(z)]\to 0 \quad \text{as } q\to\infty$$

Proof. Our proof is similar to that of convergence of Robust SA in [22]. We continue using the notation in the proof of Lemma 4. Futhermore, denote

$$\begin{aligned}&x_l^*=\Pi_{X^*}(x_l)\\&c^*(z)=f(x^*,z) \text{ for } x^*\in X^*.\end{aligned}$$

By the Projection Theorem, we have

$$||x_{l+1} - x^*_{l+1}|| \leq ||x_{l+1} - x^*_l||. \tag{17}$$

According to Lemma 4 and (17), we have

$$\begin{aligned}\mathbb{E}[||x_{l+1} - x^*_{l+1}||^2] \leq \mathbb{E}[||x_l - x^*_l||^2] - 2\gamma_l \mathbb{E}[f(x_l, z) - f(x^*_l, z)] + \gamma_l^2 M_G^2 \\ + 2\gamma_l \mathbb{E}[\delta_{N_l}(x_l, z) + \delta_{N_l}(x^*_l, z)]\end{aligned} \tag{18}$$

Let $a_l = \mathbb{E}[||x_l - x^*_l||^2]$, we have

$$a_{l+1} \leq a_l - 2\gamma_l \mathbb{E}[f(x_l, z) - c^*(z)] + 2\gamma_l^2 M_G^2 + 2\gamma_l \mathbb{E}[\delta_{N_l}(x_l, z) + \delta_{N_l}(x^*_l, z)] \tag{19}$$

Let $A_l = \sup\{\delta_{N_l}(x, z) : x \in X\}$. Since $\delta_N(\cdot, z) \geq \delta_{N+1}(\cdot, z)$ for all $N > 0$ and $N_l < N_{l+1}$, we have

$$A_l = \sup\{\delta_{N_l}(x, z) : x \in X\} \geq \sup\{\delta_{N_{l+1}}(x, z) : x \in X\} = A_{l+1} \tag{20}$$

Since $\delta_N(\cdot, z) \to 0$ uniformly on X as $N \to \infty$, we know that $A_l \to 0$ as $l \to \infty$. It follows from (19) that

$$a_{l+1} \leq a_l - 2\gamma_l \mathbb{E}[f(x_l, z) - c^*(z)] + \gamma_l^2 M_G^2 + 4\gamma_l A_l$$

which implies

$$\gamma_l \mathbb{E}[f(x_l, z) - c^*(z) - 2A_l] \leq \frac{1}{2} a_l - \frac{1}{2} a_{l+1} + \frac{1}{2}\gamma_l^2 M_G^2 \tag{21}$$

We let $1 \leq i \leq j$ and get

$$\sum_{l=i}^{j} \gamma_l [\mathbb{E}[f(x_l, z) - c^*(z)] - 2A_l] \leq \sum_{l=i}^{j} (\frac{1}{2} a_l - \frac{1}{2} a_{l+1}) + \sum_{l=i}^{j} \frac{1}{2}\gamma_l^2 M_G^2 \tag{22a}$$

$$= \frac{1}{2} a_i - \frac{1}{2} a_{j+1} + \frac{1}{2} M_G^2 \sum_{l=i}^{j} \gamma_l^2 \tag{22b}$$

$$\leq \frac{1}{2} a_i + \frac{1}{2} M_G^2 \sum_{l=i}^{j} \gamma_l^2 \tag{22c}$$

Since $A_l \geq A_{l+1}$ for any $l > 0$, we have

$$\sum_{l=i}^{j} \gamma_l A_i \geq \sum_{l=i}^{j} \gamma_l A_l \tag{23}$$

The combination of (22) and (23) implies

$$\sum_{l=i}^{j}\gamma_l\mathbb{E}[f(x_l,z)-c^*(z)]-A_i\sum_{l=i}^{j}\gamma_l\leq\frac{1}{2}a_i+\frac{1}{2}M_G^2\sum_{l=i}^{j}\gamma_l^2$$

Divide both side of the inequality above by $\sum_{l=i}^{j}\gamma_l$ and obtain

$$\frac{\sum_{l=i}^{j}\gamma_l\mathbb{E}[f(x_l,z)-c^*(z)]}{\sum_{l=i}^{j}\gamma_l}-A_i\leq\frac{\frac{1}{2}a_i+\frac{1}{2}M_G^2\sum_{l=i}^{j}\gamma_l^2}{\sum_{l=i}^{j}\gamma_l}$$

Let $v_l=\frac{\gamma_l}{\sum_{l=i}^{j}\gamma_l}$. Since $\sum_{l=i}^{j}v_l=1$ and $v_l>0$, by the convexity of $f(x,z)$ for a given z, we have

$$f(\sum_{l=i}^{j}v_lx_l,z)\leq\sum_{l=i}^{j}v_lf(x_l,z)$$

We let $x_i^j=\sum_{l=i}^{j}v_lx_l$ and get

$$\mathbb{E}[f(x_i^j,z)-f(x^*,z))]-A_i\leq\frac{\frac{1}{2}a_i+\frac{1}{2}M_G^2\sum_{l=i}^{j}\gamma_l^2}{\sum_{l=i}^{j}\gamma_l}\tag{24}$$

Compared to [22], we have an extra term, A_i here. Now we let $D_X=\max_{x\in X}||x-x_0||$ and thus $a_i=\mathbb{E}[||x_i-x_i^*||^2]\leq\mathbb{E}[(||x_i-x_0||+||x_0-x_i^*||)^2]\leq 4D_X^2$. Then it follows from (24) that

$$\mathbb{E}[f(x_i^j,z)-f(x^*,z))]\leq A_i+\frac{2D_X^2+\frac{1}{2}M_G^2\sum_{l=i}^{j}\gamma_l^2}{\sum_{l=i}^{j}\gamma_l}\tag{25}$$

Now let us take $n>0$ and $m>0$, we let $i_1=n, j_1=m+n-1$,..., $i_q=n+m\sum_{s=1}^{q-1}s$ and $j_q=n+m\sum_{s=1}^{q}s-1$. And we get $j_q-i_q=mq-1$. For $i_q\leq l\leq j_q$, we use a constant stepsize, which is $\tilde{\gamma}_q$. It follows from (25) that

$$\mathbb{E}[f(x_{i_q}^{j_q},z)-f(x^*,z))]\leq A_{i_q}+\frac{2D_X^2+\frac{1}{2}M_G^2mq\tilde{\gamma}_q^2}{mq\tilde{\gamma}_q}\tag{26}$$

We minimize the right hand side of (26) and get the minimizer below

$$\tilde{\gamma}_q=\frac{D_X}{M_G\sqrt{mq}}\tag{27}$$

(27) shows that the constant stepsize used in Robust LEON is optimized. By substituting the optimal constant stepsize into (26), we have

$$\mathbb{E}[f(x_{i_q}^{j_q},z)-c^*(z))]\leq A_{i_q}+\frac{4D_XM_G}{\sqrt{mq}}\tag{28}$$

Let $\tilde{x}_q = x_{i_q}^{j_q}$ and construct a sequence $\{\tilde{x}_q\}$ with respect to q. As $q \to \infty$, from the left hand side of the (20), we have

$$A_{i_q} \to 0 \quad \text{and} \quad \frac{4D_X M_G}{\sqrt{mq}} \to 0$$

which implies that

$$\mathbb{E}[f(\tilde{x}_q, z) - c^*(z))] \to 0$$

■

Note that A_{iq} in (28) does not depend on the trajectory of the estimated solution.

4 Robust LEON with kNN Estimation

In this section, we begin by providing the mathematical formulations of both kNN estimator of the true objective function and kNN estimator of the true subgradient and then analyze the bias of the kNN estimate. Next, we show that the bias from the kNN estimation satisfies the conditions in Theorem 1.

After Fix and Hodges Jr [11] proposed the nearest-neighbor method in 1951, several articles [6, 7, 15, 28, 29] have contributed to the consistency of kNN regression, among which Walk [28] shows that the kNN estimate converges to its true conditional expectation almost surely when $k(N) \to \infty$, $\frac{k(N)}{N} \to \infty$ as $N \to 0$ and $k(N)$ varies regularly with exponent β (e.g. $k(N) = \lfloor N^\beta \rfloor$). We will utilize the Strong Law of Large Numbers of the k^{th} nearest point (see [15] in Lemma 6.1) to analyze the bias of kNN estimator of true objective function. One bright side of kNN estimation is that model-fitting is not necessary [12]. Given a positive integer k, a dataset with size N containing $\{(z_1, w_1), (z_2, w_2),..., (z_N, w_N)\}$ and an observation of predictor whose value is z, we aim to find the k data points from this dataset closest (in distance) to z. We require that ties are broken randomly and k must be smaller than N.

Here, we provide the details of how to calculate $v_{N,i}(z)$, which corresponds to data point z_i, via kNN estimation. If z_i belongs to the set of k nearest neighbors of z, $v_{N,i}(z)$ is set to be 1; otherwise, $v_{N,i}(z)$ is set to be 0. kNN estimation is very sensitive to the choice of k and the units of measurement in the predictor space. Sometimes it is useful to do the min-max normalization on the predictors to eliminate the effect of units of measurement before kNN estimation.

We let $S_{k,N}(z)$ be the set of k nearest neighbors of z from a dataset with size N and formulate a kNN estimator of "true" problem (1) approximated by (29) below.

$$\min_{x \in X} \ \hat{f}_{k,N}(x, z) = \frac{1}{k} \sum_{i=1}^{N} F(x, W_i) \mathbb{I}(Z_i \in S_{k,N}(z)) \tag{29}$$

It has been discussed that the optimal solution of approximation problem (29) can be used as an eligible estimate of the true optimal solution [2]. However, since the runtime of naive kNN algorithm is proportional to the dimension of the vector Z and the sample size (calculating distances of N data points from a dataset consumes $O(N)$ time and sorting all the distances by using quick sort takes $O(N \log N)$ time), finding such a set of k nearest neighbors of a given point can be computationally expensive for large datasets. On the other hand, Robust LEON-kNN uses far fewer data points in each iteration and also keeps improving estimated solutions. In general, Robust LEON-kNN requires a small dataset to update the estimate when it is far from the optimal solution set and a large dataset to further improve the estimate when it is very close to the optimal solution set.

Since $v_{N,i}(z) = \frac{1}{k}\mathbb{I}(Z_i \in S_{k,N}(z)) \geq 0$, which satisfies assumption **A0**, Lemma 2 implies $\hat{f}_{k,N}(x,z)$ is convex on X for μ_Z (a.s.). The subgradient of $\hat{f}_{k,N}(x,z)$ can be written as follows.

$$\hat{G}_{k,N}(x,z) = \frac{1}{k}\sum_{i=1}^{N} G(x,W_i)\mathbb{I}(Z_i \in S_{k,N}(z)) \tag{30}$$

We refer to (30) as the kNN estimate of SQG. It is worth noting that z is fixed throughout this paper. The bound provided next will be useful for the analysis of specific non-parametric methods such as kNN and kernel estimation.

Lemma 5. *Suppose that **A0**, **A1** and **A3** - **A5** are satisfied. Then the following holds:*

$$\mathbb{E}[|f(x,Z_1) - f(x,z)|v_{N,1}(z)] \leq \mathbb{E}[C(x)||Z_1 - z||^p v_{N,1}(z)].$$

Proof. See the Appendix. ■

Lemma 6 shows that the bias of the kNN estimate satisfies conditions of Theorem 1.

Lemma 6. *Let z be fixed. Let $v_{N,i} = \frac{1}{k}\mathbb{I}(Z_i \in S_{k,N}(z))$. and **A0** - **A5** are satisfied. Further suppose that $||Z||$ is bounded almost surely (i.e. there exits $T > 0$ such that $||Z|| \leq T$ a.s.). Then the following hold:*

(1) There exists $\delta_N(x,z)$ such that $|\mathbb{E}[\hat{f}_{k,N}(x,z)] - f(x,z)| \leq \delta_N(x,z)$.

(2) $\delta_N(x',z) \to 0$, as $N \to \infty$, for any $x' \in X$.

(3) $\delta_N(\cdot,z)$ decreases monotonically with N.

(4) As $N \uparrow \infty$, $\delta_N(\cdot,z) \to 0$ uniformly on X.

Proof. See the Appendix. ■

Using Theorem 1 and Lemma 6, we are able to show that the expected optimality gap from the

sequence of estimated solutions generated by Robust LEON-kNN converges to 0. The formal corollary is provided below.

Corollary 1 (Asymptotic convergence of Robust LEON-kNN)**.** *Let z be fixed. Let $\tilde{x}_q$ be generated by Robust LEON-kNN, where q stands for the iteration number. Suppose assumptions **A0** - **A5** are satisfied. Further assume that $||Z||$ is bounded almost surely. Let l denote the l^{th} iteration of LEON Update. In each iteration of LEON update (process of generating x_l), a new dataset with larger size is generated (i.e. $N_{l+1} \geq N_l + 1$ for $l \in \mathbb{N}$). Then the following holds:*

$$\mathbb{E}[f(\tilde{x}_q, z) - c^*(z)] \to 0 \quad \text{as } q \to \infty$$

Proof. Lemma 6 implies that there exists $\delta_{N_l}(x, z)$ such that

$$|\mathbb{E}[\hat{f}_{k,N_l}(x, z)] - f(x, z)| \leq \delta_{N_l}(x, z).$$

Since $\{N_l\}_l$ is an increasing sequence and $N_l \to \infty$ as $l \to \infty$, Lemma 6 also implies that $\delta_{N_l}(\cdot, z) \to 0$ uniformly on X, as $l \to \infty$ and moreover, it also implies that $\delta_N(\cdot, z)$ is also monotonically decreasing with respect to N. Thus, all the conditions in Theorem 1 are satisfied, which completes the proof. ■

5 Robust LEON with Kernel Estimation

Since Nadaraya [21] and Watson [30] proposed the famous Nadaraya-Watson kernel estimate in 1964, it has been used in biostatistics [4], economics [3], image processing [27] etc. Several papers [13, 14, 6, 19, 15, 29] further refined the theory of kernel estimation. Walk [29] proved the strong pointwise consistency of Nadaraya-Watson kernel regression estimate under ρ mixing and α mixing. Given the range of applications they represent, we are hopeful that decision models in these areas will be able to integrate non-parametric estimation within decision models in the future. For these reasons, we extend the scope of our algorithmic procedures to a much larger domain.

Let $K(x)$ be a kernel function. Let h_N be the bandwidth, which is a smoothing parameter depending on N. The Nadaraya-Watson kernel estimator of true objective function and its subgradient are written as follows.

$$\hat{f}_N(x, z) = \frac{\sum_{i=1}^N F(x, W_i) K(\frac{z - Z_i}{h_N})}{\sum_{i=1}^N K(\frac{z - Z_i}{h_N})} \tag{31}$$

$$\hat{G}_N(x, z) = \frac{\sum_{i=1}^N G(x, W_i) K(\frac{z - Z_i}{h_N})}{\sum_{i=1}^N K(\frac{z - Z_i}{h_N})} \tag{32}$$

We shall restrict the choices to kernel functions with compact support, such as the naive kernel, Epanechikov kernel and quartic kernel. The mathematical formulations of these kernel functions are provided below [29].

Naive kernel: $K(z) = \mathbb{I}_{\{||z||\leq 1\}}$.

Epanechnikov kernel: $K(z) = (1 - ||z||^2)\mathbb{I}_{\{||z||\leq 1\}}$.

Quartic kernel: $K(z) = (1 - ||z||^2)^2\mathbb{I}_{\{||z||\leq 1\}}$.

In the literature on kernel method, h_N is known as bandwidth, and it decreases as N increases. One example is that $h_N = CN^{-\beta},\ \beta \in (0, \frac{1}{d_Z})$. For simplicity, we let $K_{N,i} = K(\frac{z-Z_i}{h_N})$ and $\mu(N) = \mathbb{E}[K_{N,1}] = \int K(\frac{z-Z_1}{h_N})d\mathbb{P}$. One difficulty associated with Nadaraya-Watson kernel estimation whose kernel has compact support, is that there is a non-zero probability that $\sum_{i=1}^N K_{N,i} = 0$, which implies that there is positive probability that both numerators and denominators of $\hat{f}_N$ are 0. In that case, the value of $\hat{f}_N$ is undefined. In other words, it is possible that there is no data point in a ball with radius h_N centered around z from a dataset with size N, which makes the kernel estimation meaningless. To overcome this situation, we propose to use a result due to [14].

Lemma 7. *Assume that the following hold:*

(1) There exists positive c and r such that $c\mathbb{I}_{\{||z||\leq r\}} \leq K(z)$. There exists $H(t)$ which is bounded, decreasing in $[0,\infty)$ and $t^dH(t) \to 0$ as $t \to \infty$.

(2) For $H(\cdot)$ in (1), there exists c_1 and c_2 such that $c_1H(||z||) \leq K(z) \leq c_2H(||z||)$.

(3) $h_N \to 0$ and $Nh_N^d \to \infty$ as $N \to \infty$.

(4) $\sum_{N=1}^\infty \exp(-\alpha N h_N^{d_Z}) < \infty$ (or $h_N = CN^{-\beta},\ \beta \in (0, \frac{1}{d_Z})$).

Then

$$\frac{\sum_{i=1}^N K_{N,i}}{N\mu(N)} \to 1 \text{ a.s.} \quad \text{as } N \to \infty$$

Proof. See the proof of case for convergence of B_{2n} in Theorem 2 [14]. ■

We let $\alpha_N = \frac{\sum_{i=1}^N K_{N,i}}{N\mu(N)}$. Since Lemma 7 [14] shows that $\frac{\sum_{i=1}^N K_{N,i}}{N\mu(N)} \to 1$ a.s., we should consider the following alternative model when analyzing the asymptotic property of Robust LEON-kernel. After replacing $\sum_{i=1}^N K_{N,i}$ by $N\mu(N)$, the estimator of true objective function can be written as follows.

$$\tilde{f}_N(x,z) := \frac{\hat{f}_N(x',z)}{\alpha_N} = \frac{\sum_{i=1}^N F(x,W_i)K_{N,i}}{N\mu(N)}$$

In practice, $N\mu(N)$ is unknown, but it can be estimated via $\sum_{i=1}^N K_{N,i}$. Similarly, a subgradient of $\tilde{f}_N(x,z)$ can be written as

$$\tilde{G}_N(x,z) = \frac{\sum_{i=1}^{N_l} G(x,W_i)K_{N_l,i}}{N\mu(N)} \tag{33}$$

It is obvious that naive kernel, Epanechnikov kernel and quartic Kernel satisfy the the conditions in Lemma 7. Note that all of the kernels mentioned above are bounded by 1, and they become 0 when z exceeds 1. More details are provided in the Appendix. We shall restrict our focus to the class of Robust LEON-kernel in which the weight function $v_{N,i}$ is calculated by $\frac{K_{N,i}}{N\mu(N)}$.

The following lemma shows that the bias of $\tilde{f}_N(x,z)$ satisfies the condition in Theorem 1.

Lemma 8. *Let z be fixed. Suppose that $h_N = CN^{-\beta}(\beta \in (0, \frac{1}{d_Z}))$, where d_Z is the dimension of Z, and $v_{N,i}$ is calculated by $\frac{K_{N,i}}{N\mu(N)}$. Further suppose that K is naive kernel, Epanechikov kernel or quartic kernel, **A0** - **A5** are satisfied. Then the following holds:*

(1) There exists $\delta_N(x,z)$ such that $|\mathbb{E}[\tilde{f}_N(x,z)] - f(x,z)| \leq \delta_N(x,z)$.

(2) $\delta_N(\cdot,z)$ is monotonically decreasing with respect to N

(3) As $N \uparrow \infty$, $\delta_N(\cdot,z) \to 0$ uniformly on X.

Proof. See the Appendix. ■

As for Lemma 8, **A5** can be relaxed to that there exists $r_z > 0$ such that $|f(x,z_1) - f(x,z_2)| \leq C(x)||z_1 - z_2||^p$, for every $x \in X$ and $||z_1 - z_2|| \leq r_z$. That is, the inequality in **A5** can only be required to be locally true. The reason is that there exists a finite N^* such that the bandwidth, h_N, will be smaller than r_z (i.e. $h_N \leq r_z$ for all $N \geq N^*$), which implies that Lemma 8 will hold for large enough N. We end this section with the following result which uses both Lemma 8 and Theorem 1 to show that the expected optimality gap from the sequence of estimated solutions generated by Robust LEON-kernel converges to 0.

Corollary 2 (Asymptotic convergence of Robust LEON-kernel)**.** *Let z be fixed. Let $\tilde{x}_q$ be generated by Robust LEON-kNN/LEON-kernel, where q stands for the iteration number. Suppose assumptions **A0** - **A5** are satisfied. Let l denote the l^{th} iteration of LEON Update. In each iteration of LEON update (process of generating x_l), a new dataset with larger size is generated (i.e. $N_{l+1} \geq N_l + 1$ for $l \in \mathbb{N}$). Then the following holds:*

$$\mathbb{E}[f(\tilde{x}_q,z) - c^*(z)] \to 0 \quad \text{as } q \to \infty$$

Proof. The proof is similar to the proof of Corollary 1. See the Appendix. ■

6 Numerical Experiments

One practical example of the problems of interest is a two-stage stochastic linear programming, whose mathematical formulation is given below.

$$\begin{aligned} \min_{x} \; & c_1^\top x + \mathbb{E}[Q(x, W)|Z = z] \\ \text{s.t. } & A_1 x = b_1 \\ & x \geq 0 \end{aligned}$$

where $Q(x, W)$ depends on the realization of Z, and it is the optimal cost of the following subproblem

$$\begin{aligned} \min_{y} \; & c_2^\top y \\ \text{s.t. } & A_2 y + B_2 x = b_2(W) \\ & x \geq 0 \end{aligned}$$

Here, c_1, c_2 and b_1 are deterministic column vectors. A_1, A_2 and B_2 are deterministic matrices. $b_2(W)$ is stochastic column vector which depends on W. To draw a connection with the standard problem (1), note that $F(x, W) = c_1^\top x + Q(x, W)$, $f(x, z) = c_1^\top x + \mathbb{E}[Q(x, W)|Z = z]$, and $X = \{x : A_1 x = b_1, x \geq 0\}$.

6.1 A Two-Stage Shipment Planning Problem

In this section, we analyze the computational performance of Robust LEON by solving a two-stage shipment planning problem introduced by Bertsimas and Kallus [2]. Here is a brief introduction of this shipment planning problem. In the first stage, we have 4 warehouses which allow us to store the products at the cost of \$5 before they are sold in the second stage. In the second stage, the stored products in the 4 warehouses are shipped to 12 locations to satisfy each individual demands. The cost of shipment from one warehouse to one location is proportional to their distance. We are allowed to make last-minute replenishment from the 4 warehouses at a cost of \$100 to satisfy extra demands. The goal of this problem is to decide the number of products stored in each of the 4 warehouses in the first stage to minimize the total costs.

We also follow the procedure in [2] to generate predictor and response pairs. The responses are the demands, and predictors are 3 dimensional ARMA(2,2) time series and responses are generated by the linear transformation of corresponding predictors and standard normal errors. We elaborate on scenario generation in the l^{th} iteration below.

1. Inputs: α_0 and β are positive integers. I_{start} and I_{end} are start index and end index in the last iteration, respectively. l is iteration number. For $l = 1$, Z_1, Z_2, U_1 and U_2 are the initial points. For $l > 1$, $Z_{I_{start}-1}, Z_{I_{start}-2}, U_{I_{start}-1}$ and $U_{I_{start}-2}$ are the initial points. (See equation (3) for the definition of U)

2. Set $I_{start} \leftarrow I_{start} + \alpha_0 + \beta(l-2)$, and $I_{end} \leftarrow I_{end} + \alpha_0 + \beta(l-1)$. Generate a sequence of 3 dimensional ARMA(2,2) time series and its response, $\{(Z_{I_{start}}, W_{I_{start}}), ..., (Z_{I_{end}}, W_{I_{end}})\}$, which is the dataset used in the l^{th} iteration.

We provide an illustration of two approaches used to measure solution quality below. In the first approach (*validation approach I*), which is introduced by Bertsimas and Kallus [2], the performance (true cost of estimated solution) is estimated by the average cost of one hundred thousand scenarios from the true conditional distribution given the observed predictor, z, (as shown in equation (5)). For the mathematical formulation of calculating this metric, please see equation (6).

In the second approach (*validation approach II*), we strive to use kNN to calculate the estimated true cost. In particular, one thousand nearest neighbors of observed z from a validation dataset with size one million are generated as candidate scenarios (as shown in equation (4)). The performance of a given estimated solution is quantified as the average cost of these 1000 candidate scenarios by using the same solution. We let $(\bar{Z}_i, \bar{W}_i)$ denote a sample from the validation set, $\bar{S}_N = \{(\bar{Z}_1, \bar{W}_1), ..., (\bar{Z}_N, \bar{W}_N)\}$, generated by unconditional distribution. Let $\bar{S}_{k,N}(z)$ be the set of k nearest neighbors of z from $\bar{S}_N$. The estimated true cost of a candidate solution and the kNN estimate of the true optimal cost are written below.

$$\bar{\theta}(\hat{x}) = \frac{1}{k} \sum_{i=1}^{N} \mathbb{I}(\bar{Z}_i \in \bar{S}_{k,N}(z)) F(\hat{x}, \bar{W}_i) \tag{34}$$

$$\theta^*_{k\text{NN}} = \min_{x \in X} \frac{1}{k} \sum_{i=1}^{N} \mathbb{I}(\bar{Z}_i \in \bar{S}_{k,N}(z)) F(x, \bar{W}_i) \tag{35}$$

Next, we provide the specific algorithmic parameters, including parameters (e.g. bandwidth) for non-parametric estimation as well as the ones for optimization. We consider the case in which the observed predictor is $[-0.3626, 0.5871, -0.2987]$. The initial point of x is $[20, 20, 20, 20]$. As for Robust LEON, we set $D_x = 40$, $M = 30$ and $m = 5$. Initial batch size is set to 100 and the increment on the batch size per iteration is 10. The parameter for kNN is $k = 20$, while parameters for kernel estimation are $C = 1$, $n_z = 3$ (dimension of the predictor), $\beta = \frac{0.2}{n_z}$ and bandwidth $h_N = CN^{-\beta}$. Iteration number of Robust

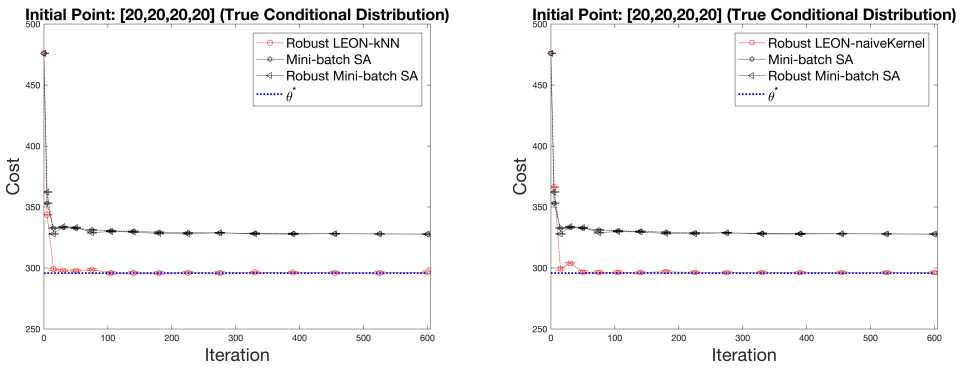
LEON in the following graphs is the iteration number of LEON updates. For the purpose of comparison, stochastic subgradient in the l^{th} iteration from Mini-batch SA or Mini-batch Robust SA is calculated by the average of all the scenarios of the dataset generated in the l^{th} iteration. The initial step size for mini-bacth SA is 0.8.

Table 1: Computational results of the first setup

Algorithm	L	Obj(I)	Obj(II)
Mini-batch SA	600	327.7690($\pm$0.4190)	328.1570($\pm$4.8240)
Robust Mini-batch SA	600	327.8040($\pm$0.4290)	328.2600($\pm$4.9540)
Robust LEON-kNN	600	296.5420($\pm$0.9180)	298.7950($\pm$10.8370)
Robust LEON-naiveKernel	600	296.1540($\pm$1.0230)	298.2510($\pm$11.9000)
Robust LEON-EpanechnikovKernel	600	295.9950($\pm$1.0200)	298.0750($\pm$11.8610)
Robust LEON-quarticKernel	600	295.9280($\pm$1.0190)	297.9960($\pm$11.8600)

[a.] Observed predictor is $[-0.3626, 0.5871, -0.2987]$. Initial point is $[20, 20, 20, 20]$. Initial stepsize is 0.8. Initial batch size is 100. Increment on the batch size per iteration is 10. Parameter for kNN is $k = 20$. Parameters for kernel estimation are $C = 1$, $n_z = 3$ (dimension of the predictor), $\beta = \frac{0.2}{n_z}$ and bandwidth $h_N = CN^{-\beta}$. Parameters for Robust LEON are $D_x = 40$, $M = 30$, $n = 1$ and $m = 5$. $\theta^* = 295.792$ in *validation approach I* and $\theta^*_{kNN} = 297.748$ in *validation approach II*. L is number of iterations in Mini-batch SA. L is the total number of times calling LEON Update Rule in Robust LEON. L is the total number of times calling mini-batch SA in Robust Mini-batch SA.

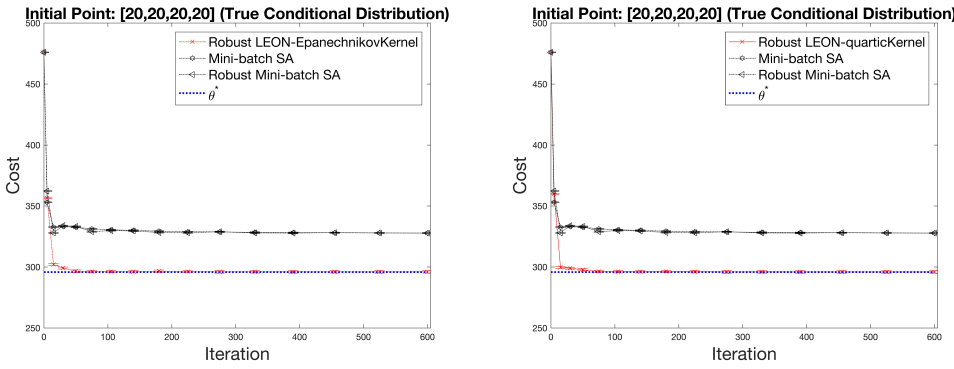
Figure 2: Computations of Robust LEON-kNN and Robust LEON-naiveKernel (*Validation Approach I*)



Initial point is [20,20,20,20], and initial step size is 0.8.

The data in Table 1 shows the solution quality (the lower the value, the better the solution quality) of each algorithm after 600 iterations. Figures 2 and 3 show the computational progress of each algorithm in terms of solution quality based on *validation approach I* introduced earlier in this section. The dotted lines in Figures 2 and 3 are plotted according to (7), which is the estimated optimal cost. The vertical bar on each point shows the 95% confidence interval of the corresponding estimated cost. For the reader interested in the graphs of solution quality evaluated by *validation approach II*, please see the Appendix. Basically, the converging trends shown in graphs from both types of validation approaches are similar. It is worth noting that the interval shown in the column of Obj(II) from the tables suggest that the

Figure 3: Computations of Robust LEON-EpanechnikovKernel and Robust LEON-quarticKernel (*Validation Approach I.*



Initial point is [20,20,20,20], and initial step size is 0.8

deviation of kNN point estimate may not be a rigorous 95% confidence interval, whereas the interval in the column of Obj(I) shows the 95% confidence interval. The main difference of two approaches is that *validation approach II* has larger deviation than *validation approach I*, which can also be verified in Table 1. In particular, Figure 2 and Figure 3 show that the optimality gap of (Robust) Mini-batch SA converges to a positive number while the optimality gap of the Robust LEON converges to 0.

For the readers who are interested in changing the initial setup of Robust LEON, please see the Appendix. The same two-stage shipment planning problem is also used to numerically analyze the bias of kNN estimator, where we replicate the experiments 60 times. The bias analysis in the appendix sheds some light on the claim the bias of kNN estimator decreases as the sample size increases.

7 Conclusions

Bertsimas and Kallus [2] provide the formulation and consistency analysis of problem (1). For algorithmic purposes, they recommend using any SA algorithm for solving an approximation (i.e., (2)). In such an implementation, the SA algorithm does not encounter a randomized but a deterministic function as in (2). In contrast, for each LEON update of Robust LEON algorithm, the functional estimate is updated based on a new dataset. Thus, our approach adopts a simultaneous estimation-optimization process, whereas the recommendation of [2] separates the estimation from optimization.

Under mild assumptions, such as probabilistic existence of $Z = z$, compact and convex feasible set, convex cost function, and the i.i.d. assumption for the predictor-response pairs, we have shown the convexity of the true objective function, and its non-parametric estimator (Lemma 2). Next, Lemma 2 and Lemma 4 (sufficient reduction in LEON Update Rule) are used to prove the asymptotic convergence of Robust LEON (Theorem 1), which is the mathematical foundation of Robust LEON-kNN and Robust

LEON-kernel. Since the entire framework is parametrized by z, it needs to be highlighted that the convergence of Robust LEON is consistent with μ_Z almost every z. With further assumption on the Lipschitz-type continuity of the true objective function with respect to the predictor (**A5**), we prove the asymptotic convergence of both Robust LEON-kNN (Corollary 1) and Robust LEON-kernel (Corollary 2). Finally, computational results in a two-stage shipment planning problem confirm the conclusions of Robust LEON.

Despite some of the advances suggested in this paper, there are several challenges that remain. One challenge is that in-sample stopping rule of Robust LEON algorithm is under study. Another one is that the universal performance of Robust LEON for all the possible realizations of the predictor is unknown. One future direction can be the analysis of the rate of convergence of LEON in the context of its universal performance. It is still unknown whether the bias of kernel estimator via kernel function with noncompact support (e.g., Gaussian kernel), converges uniformly on a bounded compact set. Besides, when a true subgradient is estimated by the Nadaraya-Watson kernel weighted average, direct proof of asymptotic convergence of such class of LEON-kernel is still under study.

While kernel approximations have been used to approximate dynamic programming value functions, we are not aware of SP algorithms which have imported kernel and other non-parametric estimation as an integral part of the solution algorithm. We expect this combination of estimation and optimization to be the fundamental step for distribution-free algorithms in SP.

Acknowledgements

We thank Prof. Phebe Vayanos for her participation in early discussion of this research. This research was supported by grants from NSF-1822327 and AFOSR FA9550-20-1-0006.

Appendix

Mini-batch SA

1. Choose a diminishing step size rule $\{\gamma_l\}_l$, the maximum number of iterations (l_{max}), batch size (N), initial points of predictor (Z_1 and Z_2), and corresponding U_1 and U_2 in the 3-dimensional ARMA(2,2) time series. Set the iteration number $l \leftarrow 0$, the start index $I_{start} \leftarrow 1$ and the end index $I_{end} \leftarrow N$.

2. Set $l \leftarrow l + 1$. Generate a sequence of time series and its response,

$\{(Z_{I_{start}}, W_{I_{start}}), ..., (Z_{I_{end}}, W_{I_{end}})\}$. (Note: For $l = 1$, the initial points are Z_1, Z_2, U_1 and U_2. For $l > 1$, the initial points are $Z_{I_{start}-1}, Z_{I_{start}-2}, U_{I_{start}-1}$ and $U_{I_{start}-2}$).

3. Calculate the stochastic subgradient, $\bar{G}_l(x_l) = \frac{1}{N}\sum_{i=I_{start}}^{I_{end}} G(x_l, W_i)$.

4. Update the estimated solution, $x_{l+1} = \Pi_X(x_l - \gamma_l \bar{G}_l(x_l))$, where $\Pi_X(\cdot)$ denotes the standard projection operator.

5. If $l \geq l_{max}$, stop and output the estimated solution. Otherwise, set $I_{start} \leftarrow I_{start} + N$, and $I_{end} \leftarrow I_{end} + N$ and repeat from step 2.

Example of Lipschitz-type continuous function

Let W and Z be one dimensional random variables. Suppose that $|Z| \leq a$ a.s. and $W = Z + \epsilon$, where ϵ is standard normal random variable. For $|z| \leq a$ we want to solve the following problem:

$$\min_x f(x, z) = \mathbb{E}[(x - W)^2 | Z = z].$$

Here, the objective function can be written as

$$\begin{aligned} f(x, z) &= \mathbb{E}[(x - z - \epsilon)^2] \\ &= 1 + (x - z)^2 \end{aligned}$$

Here, $\epsilon - x + z$ is normal random variable with mean $-x + z$ and variance 1. For $|z_1| \leq a, |z_2| \leq a$ we have

$$\begin{aligned} |f(x, z_1) - f(x, z_2)| &\leq |(x - z_1)^2 - (x - z_2)^2| \\ &\leq |z_1 - z_2||2x - z_1 - z_2| \\ &\leq (|2x| + 2a)|z_1 - z_2| \end{aligned}$$

Thus, for this example, $C(x) = 2(|x| + a)$. Furthermore, if the feasible region, X, of x is compact, then $f(x, z)$ is Hödler continuous in z. (For $|z_1| \leq a, |z_2| \leq a$, there exists C such that $|f(x, z_1) - f(x, z_2)| \leq C|z_1 - z_2|$ for all $x \in X$.)

Proof of Lemma 2

Proof. Proof of (1). Let $x_1, x_2 \in X$ and $\lambda \in [0, 1]$. Since $F(\cdot, W)$ is convex on X a.s., we have

$$F(\lambda x_1 + (1 - \lambda)x_2, W) \leq \lambda F(x_1, W) + (1 - \lambda)F(x_2, W) \text{ a.s.}, \tag{36}$$

Based on Lemma 1, we have

$$\mathbb{E}[F(\lambda x_1 + (1-\lambda)x_2, W)|Z = z] \leq \mathbb{E}[\lambda F(x_1, W) + (1-\lambda)F(x_2, W)|Z = z]. \tag{37}$$

Since $f(\lambda x_1 + (1-\lambda)x_2, z) = \mathbb{E}[F(\lambda x_1 + (1-\lambda)x_2, W)|Z = z]$ and $\lambda f(x_1, z) + (1-\lambda)f(x_2, z) = \mathbb{E}[\lambda F(x_1, W) + (1-\lambda)F(x_2, W)|Z = z]$, it follows from (37) that

$$f(\lambda x_1 + (1-\lambda)x_2, z) \leq \lambda f(x_1, z) + (1-\lambda)f(x_2, z).$$

Since $\lambda \in [0, 1]$ is arbitrary, it shows that $f(\cdot, z)$ is convex on X for a given $z \in \mathbb{R}^{n_z}$. ■

Proof. Proof of (2). Since $F(\cdot, w)$ is convex on X for any $w \in supp(W)$, we have that, for $x_1, x_2 \in X$ and $i \in \{1, \ldots, N\}$, (36) holds for all W_i and $\lambda \in [0, 1]$. Since $v_{N,i}(z) \geq 0$ for all (N, i), it follows that

$$v_{N,i}F(\lambda x_1 + (1-\lambda)x_2, W_i) \leq \lambda v_{N,i}F(x_1, W_i) + (1-\lambda)v_{N,i}F(x_2, W_i) \tag{38}$$

By the definition of $\hat{f}_N(x, z)$, summing (38) over all indices, i, yields

$$\hat{f}_N(\lambda x_1 + (1-\lambda)x_2, z) \leq \lambda \hat{f}_N(x_1, z) + (1-\lambda)\hat{f}(x_2, z)$$

Since λ is arbitrary, this completes the proof. ■

Proof of Lemma 3

Proof. Let $x, x^* \in X$, where x^* denotes an optimal solution. By assumption, we have

$$\mathbb{E}[\hat{f}_N(x, z)] \geq f(x, z) - \delta_N(x, z), \tag{39}$$

$$\mathbb{E}[\hat{f}_N(x^*, z)] \leq f(x^*, z) + \delta_N(x^*, z). \tag{40}$$

By the convexity of $\hat{f}_N(\cdot, z)$ with respect to x for a given z, we have

$$\hat{f}_N(x^*, z) - \hat{f}_N(x, z) \geq \langle \hat{G}_N(x, z), x^* - x \rangle.$$

By taking the expectation of the both sides of the equation above, we get

$$\mathbb{E}[\hat{f}_N(x^*, z)] - \mathbb{E}[\hat{f}_N(x, z)] \geq \langle \mathbb{E}[\hat{G}_N(x, z)], x^* - x \rangle \tag{41}$$

Combining equations (39), (40) and (41), we obtain

$$f(x^*,z)+\delta_N(x^*,z)-[f(x,z)-\delta_N(x,z)]\geq\langle\mathbb{E}[\hat{G}_N(x,z)],x^*-x\rangle,$$

which implies that

$$f(x^*,z)-f(x,z)\geq\langle\mathbb{E}[\hat{G}_N(x,z)],x^*-x\rangle+(-\delta_N(x^*,z)-\delta_N(x,z))$$

Let $\tau_N=-\delta_N(x^*,z)-\delta_N(x,z)$. Since $\delta_N(x,z)$ converges uniformly to 0 on X as $N\to\infty$, we conclude that $\tau_N\to 0$ as $N\to\infty$. By Definition 1, it shows that $\hat{G}_N(x,z)$ is the stochastic quasi-gradient of $f(x,z)$. ■

Proof of Lemma 5

Proof. Let $H=\{\omega:|f(x,Z_1(\omega))-f(x,z)|v_{N,1}(z)\leq C(x)||Z_1(\omega)-z||^p\}$. Then based on assumption **A4**, we have $\mathbb{E}[\mathbb{I}(Z_1\in H)]=\mathbb{P}(H)=1$. $\mathbb{E}[|F(x,W)|]<\infty$, implies that there exists $M_F<\infty$ such that $\mathbb{E}[|F(x,W)||Z]<M_F$ for μ_Z (a.s.). Hence, we have

$$f(x,Z)=\mathbb{E}[F(x,W)|Z]\leq\mathbb{E}[|F(x,W)||Z]<M_F$$

Consequently,

$$\begin{aligned}\mathbb{E}[|f(x,Z_1)-f(x,z)|v_{N,1}(z)]&\leq\mathbb{E}[|f(x,Z_1)-f(x,z)|v_{N,1}(z)\mathbb{I}(Z_1\in H)]\\&\quad+\mathbb{E}[|f(x,Z_1)-f(x,z)|v_{N,1}(z)\mathbb{I}(Z_1\notin H)] \quad (42a)\\&\leq\mathbb{E}[C(x)||Z_1-z||^p v_{N,1}(z)\mathbb{I}(Z_1\in H)]\\&\quad+2M_F\mathbb{E}[\mathbb{I}(Z_1\notin H)] \quad (42b)\end{aligned}$$

Since $\mathbb{E}[\mathbb{I}(Z_1\notin H)]=\mathbb{P}(\mathbb{I}(Z_1\notin H))=0$ and $\mathbb{E}[C(x)||Z_1-z||^p v_{N,1}(z)\mathbb{I}(Z_1\in H)]\leq\mathbb{E}[C(x)||Z_1-z||^p v_{N,1}(z)]$, it follows from (42b) that $\mathbb{E}[|f(x,Z_1)-f(x,z)|v_{N,1}(z)]\leq\mathbb{E}[C(x)||Z_1-z||^p v_{N,1}(z)]$. ■

Proof of Lemma 6

Proof. Proof of (1). Let $\mathcal{S}=\sigma\{Z_i:i\in 1,2,...,N\}$ denote the σ-algebra generated by $\{Z_i:i\in 1,2,...,N\}$. Since (Z_1,W_1),...,(Z_N,W_N) are independent and identically distributed, the expectation of

k-NN estimate in (29) can be decomposed into the following form [28]:

$$\begin{aligned}\mathbb{E}[\hat{f}_{k,N}(x,z)] &= \frac{1}{k}\sum_{i=1}^{N}\mathbb{E}[\mathbb{E}[F(x,W_i)\mathbb{I}(Z_i \in S_{k,N}(z))|\mathcal{S}]] \\ &= \frac{1}{k}\sum_{i=1}^{N}\mathbb{E}[\mathbb{E}[F(x,W_i)|Z_i]\mathbb{I}(Z_i \in S_{k,N}(z))] \\ &= \frac{N}{k}\mathbb{E}[\mathbb{E}[F(x,W)|Z_1]\mathbb{I}(Z_1 \in S_{k,N}(z))]] \\ &= \frac{N}{k}\mathbb{E}[f(x,Z_1)\mathbb{I}(Z_1 \in S_{k,N}(z))]\end{aligned} \tag{43}$$

On the other hand, we have $\mathbb{E}[\mathbb{I}(Z_1 \in S_{k,N}(z))] = \mathbb{P}(Z_1 \in S_{k,N}(z)) = 1 - \frac{\binom{N-1}{k}}{\binom{N}{k}} = \frac{k}{N}$, because Z_1,...,Z_N are equally likely to be in the set of k nearest neighbors of z. Hence, it implies that for any constant $a \in \mathbb{R}$, the following equality holds.

$$a = \frac{N}{k}\mathbb{E}[a\mathbb{I}(Z_1 \in S_{k,N}(z))] \tag{44}$$

By using equation (43) and (44), we attain an upper bound of $|\mathbb{E}[\hat{f}_{k,N}(x,z)] - f(x,z)|$ below,

$$\begin{aligned}|\mathbb{E}[\hat{f}_{k,N}(x,z)] - f(x,z)| &\leq \frac{N}{k}\mathbb{E}[|f(x,Z_1) - f(x,z)|\mathbb{I}(Z_1 \in S_{k,N}(z))] \\ &\leq \frac{N}{k}C(x)\mathbb{E}[||Z_1 - z||^p\mathbb{I}(Z_1 \in S_{k,N}(z))]\end{aligned} \tag{45}$$

Defining $\delta_N(x,z) := \frac{N}{k}C(x)\mathbb{E}[||Z_1 - z||^p\mathbb{I}(Z_1 \in S_{k,N}(z))]$, (45) simplifies to

$$|\mathbb{E}[\hat{f}_{k,N}(x,z)] - f(x,z)| \leq \delta_N(x,z) \tag{46}$$

■

Proof. Proof of (2). Let $Z^N_{[k]}(z)$ be the k^{th} nearest neighbor of z in the dataset with size N. By the SLLN of kNN (Lemma 6.1 in [15]), we have

$$||Z^N_{[k]}(z) - z|| \to 0 \ a.s. \quad \text{as } N \to \infty.$$

which implies that

$$||Z^N_{[k]}(z) - z||^p \to 0 \ a.s. \quad \text{as } N \to \infty. \tag{47}$$

Equation (47) can be generalized to

$$||Z^N_{[i]}(z) - z||^p \to 0 \ a.s. \quad \text{as } N \to \infty \quad \text{for } i \leq k$$

Since k is finite, we have

$$\frac{1}{k}\sum_{i=1}^{k}||Z_{[i]}^{N}(z)-z||^{p}\to 0\ a.s.\quad \text{as } N\to\infty \tag{48}$$

which implies that

$$\frac{1}{k}\sum_{i=1}^{N}||Z_{i}^{N}(z)-z||^{p}\mathbb{I}(Z_i\in S_{k,N}(z))\to 0\ a.s.\quad \text{as } N\to\infty. \tag{49}$$

Since $||Z||\le T<\infty$ a.s., $||Z-z||^{p}\le(2T)^{p}<\infty$ a.s.. By the Dominated Convergence Theorem (or Bounded Convergence Theorem), it follows from (49) that

$$\mathbb{E}[||Z_{[i]}^{N}(z)-z||^{p}]\to 0\quad \text{as } N\to\infty\quad \text{for } i\le k, \tag{50}$$

and equation (50) implies that

$$\mathbb{E}[\frac{1}{k}\sum_{i=1}^{k}||Z_{[i]}^{N}(z)-z||^{p}]\to 0\quad \text{as } N\to\infty\quad \text{for } i\le k \tag{51}$$

Since $C(x)$ is finite, for $x\in X$, (51) implies that

$$\mathbb{E}[\frac{1}{k}C(x')\sum_{i=1}^{N}||Z_i-z||^{p}\mathbb{I}(Z_i\in S_{k,N}(z))]\to 0\quad \text{as } N\to\infty \tag{52}$$

Since $Z_1^l, Z_2^l, ..., Z_{N_l}^l$ are independent and identically distributed, we have

$$\begin{aligned}&\mathbb{E}[\frac{1}{k}C(x')\sum_{i=1}^{N}||Z_i-z||^{p}\mathbb{I}(Z_i\in S_{k,N}(z))]\\ =\ &\frac{N}{k}C(x')\mathbb{E}[||Z_1-z||^{p}\mathbb{I}(Z_1\in S_{k,N}(z))]=\delta_N(x',z)\end{aligned} \tag{53}$$

The combination of (52) and (53) implies that $\delta_N(x',z)\to 0$ as $N\to\infty$.

■

Proof. Proof of (3). We shall show that $\delta_N\ge\delta_{N+1}$ pointwise for $N>k$. Let $\{Z_1,Z_2,...,Z_N,Z_{N+1}\}$ be $N+1$ independent copies of Z. Let $Z_{[i]}^{N+1}(z)$ be the i^{th} nearest neighbor of z from $\{Z_1,Z_2,...,Z_N,Z_{N+1}\}$. Let $Z_{[i]}^{N}(z)$ be the i^{th} nearest neighbor of z from $\{Z_1,Z_2,...,Z_N\}$. It is obvious that $||Z_{[i]}^{N+1}(z)-z||^{p}\le||Z_{[i]}^{N}(z)-z||^{p}$, which implies

$$\mathbb{E}[||Z_{[i]}^{N+1}(z)-z||^{p}]\le\mathbb{E}[||Z_{[i]}^{N}(z)-z||^{p}] \tag{54}$$

Since (54) holds for $i \in \{1, 2, 3, ..., k\}$, it implies that

$$\sum_{i=1}^{k} \mathbb{E}[||Z_{[i]}^{N+1}(z) - z||^p] \leq \sum_{i=1}^{k} \mathbb{E}[||Z_{[i]}^{N}(z) - z||^p] \tag{55}$$

We also have

$$\begin{aligned} \delta_N(x, z) &= \mathbb{E}[\frac{1}{k} \sum_{i=1}^{N} C(x)||Z_i - z||^p \mathbb{I}(Z_i \in S_{k,N}(z))] \\ &= C(x) \sum_{i=1}^{k} \mathbb{E}[||Z_{[i]}^{N}(z) - z||^p] \end{aligned} \tag{56}$$

and

$$\begin{aligned} \delta_{N+1}(x, z) &= \mathbb{E}[\frac{1}{k} \sum_{i=1}^{N+1} C(x)||Z_i - z||^p \mathbb{I}(Z_i \in S_{k,N+1}(z))] \\ &= C(x) \sum_{i=1}^{k} \mathbb{E}[||Z_{[i]}^{N+1}(z) - z||^p] \end{aligned} \tag{57}$$

So the combination of (55), (56) and (57) implies $\delta_N(x, z) \geq \delta_{N+1}(x, z)$.

It is worth noting that if we generate $Z_1', Z_2', Z_3', ..., Z_N'$, which is another set of N independent copies of Z. We let $S_{k,N}'(z)$ be the kNN set of z from this new set. Then

$$\begin{aligned} \mathbb{E}[\frac{1}{k} \sum_{i=1}^{N} C(x)||Z_i' - z||^p \mathbb{I}(Z_i' \in S_{k,N}'(z))] &= \mathbb{E}[\frac{1}{k} \sum_{i=1}^{N} C(x)||Z_i - z||^p \mathbb{I}(Z_i \in S_{k,N}(z))] \\ &= \delta_N(x, z) \end{aligned}$$

This argument is very useful in understanding dynamics of our algorithm. ■

Proof. Proof of (4). According to (2) and (3) from Lemma 6, we know that $\delta_N(\cdot, z) \to 0$ pointwise on X and $\delta_N(\cdot, z)$ is monotonically decreasing with respect to N. Since $C(x)$ is continuous on X, it is obvious that $\delta_N(\cdot, z)$ is continuous on X. By Dini's Theorem [17], $\delta_N(\cdot, z)$ converges uniformly to 0 on X, as $N \to \infty$. ■

Kernel properties

For naive kernel, putting $K(z) = \mathbb{I}_{\{||z|| \leq 1\}}$, we let $H(t) = \mathbb{I}_{\{t \leq 1\}}$. Then we can choose $c_1 = 0.9$ and $c_2 = 1.1$, and condition (2) is satisfied. When $t > 1$, $H(t) = 0$, which implies that $\lim_{t \to \infty} tH(t) = 0$. Thus, $H(t)$ is nonincreasing, and the second part of condition (1) is also satisfied. As for the first part condition (1), we can choose $r = 1$ and $c = 1$.

For Epanechnikov kernel, $K(z) = (1 - ||z||^2)\mathbb{I}_{\{||z|| \leq 1\}}$, we can let $H(t) = (1 - t^2)\mathbb{I}_{\{t \leq 1\}}$. Then we can pick $c_1 = 0.9$ and $c_2 = 1.1$. Then condition (2) is satisfied. When $t > 1$, $H(t) = 0$, which implies that $\lim_{t \to \infty} tH(t) = 0$. Again, $H(t)$ is nonincreasing, and the second part of condition (1) is also satisfied.

As for the first part of condition (1), we can choose $r = 0.5$ and $c = 1 - 0.5^2 = 0.75$.

For quartic kernel, $K(z) = (1 - ||z||^2)^2 \mathbb{I}_{\{||z|| \leq 1\}}$, we can let $H(t) = (1 - t^2)^2 \mathbb{I}_{\{t \leq\}}$. We can choose $c_1 = 0.9$ and $c_2 = 1.1$. Then condition (2) is satisfied. When $t > 1$, $H(t) = 0$, which implies that $\lim_{t \to \infty} tH(t) = 0$, and $H(t)$ is nonincreasing. So the second part of condition (1) is also satisfied. As for the first part of condition (1), we can choose $r = 0.5$ and $c = (1 - 0.5^2)^2 = 0.5625$.

Proof of Lemma 8

Proof. Let $\mathcal{S}$ be the sigma algebra generated by Z_1, Z_2, ...,Z_N. Then

$$\mathbb{E}[\sum_{i=1}^{N} F(x, W_i) K_{N,i})] = \sum_{i=1}^{N} \mathbb{E}[\mathbb{E}[F(x, W_i) K_{N,i} | \mathcal{S}]] \tag{58a}$$

$$= N\mathbb{E}[\mathbb{E}[F(x, W_1)|\mathcal{S}] K_{N,1}] \tag{58b}$$

$$= N\mathbb{E}[\mathbb{E}[F(x, W_1)|Z_1] K_{N,1}] \tag{58c}$$

$$= N\mathbb{E}[f(x, Z_1) K_{N,1}]. \tag{58d}$$

In the above sequence of equations, (58b) is true, because $K_{N,1}$ is measurable on $\mathcal{S}$. Since $Z_1, Z_2, ..., Z_N$ are independent, we have $\mathbb{E}[F(x, W_1)|\mathcal{S}] = \mathbb{E}[F(x, W_1)|Z_1, Z_2, ..., Z_N] = \mathbb{E}[F(x, W_1)|Z_1]$. Since $f(x, z)$ is a constant for given x and z, we have

$$\mathbb{E}[f(x, z) K_{N,1}] = f(x, z)\mu(N) \tag{59}$$

where $\mu(N)$ is defined as $\mu(N) = \mathbb{E}[K_{N,1}]$. The combination of (58) and (59) implies that

$$\begin{aligned} |\mathbb{E}[\tilde{f}_N(x, z)] - f(x, z)| &= |\frac{\mathbb{E}[f(x, Z_1) K_{N,1}]}{\mu(N)} - f(x, z)| \\ &= \frac{1}{\mu(N)} |\mathbb{E}[(f(x, Z_1) - f(x, z)) K_{N,1}]| \\ &\leq \frac{1}{\mu(N)} \mathbb{E}[|f(x, Z_1) - f(x, z)| K_{N,1}] \end{aligned} \tag{60}$$

By Assumption **A5**, it follows from (60) that

$$\frac{1}{\mu(N)} \mathbb{E}[|f(x, Z_1) - f(x, z)| K_{N,1}] \leq \frac{1}{\mu(N)} C(x) \mathbb{E}[||Z_1 - z||^p K_{N,1}] \tag{61}$$

By the definition of naive kernel, Epanechikov kernel or quartic kernel, we have $K_{N,1} = 0$ if $||Z_1 - z|| \geq h_N$. So we obtain

$$||Z_1 - z||^p K_{N,1} = \begin{cases} ||Z_1 - z||^p K_{N,1} & \text{if } ||Z_1 - z|| \leq h_N \\ 0 & \text{otherwise} \end{cases} \tag{62}$$

(62) implies that $||Z_1 - z||^p K_{N,1} \leq h_N^p K_{N,1}$. Hence, it follows from (61) that

$$\begin{aligned} \frac{1}{\mu(N)} C(x)\mathbb{E}[||Z_1 - z||^p K_{N,1}] &\leq \frac{1}{\mu(N)} C(x) h_N^p \mathbb{E}[K_{N,1}] \\ &= C(x) h_N^p \end{aligned}$$

We let $\delta_N(x, z) = C(x)h_N^p$ and have

$$|\mathbb{E}[\tilde{f}_N(x, z)] - f(x, z)| \leq \delta_N(x, z) \tag{63}$$

For fixed x,

$$\delta_N(x, z) = C(x)h_N^p \to 0 \text{ as } N \to \infty$$

$N \leq N'$ implies that $h_N \geq h_{N'}$, which further implies that

$$\delta_N(x, z) \geq \delta_{N'}(x, z)$$

Since $C(x)$ is continuous, $\delta_N(\cdot, z)$ is also continuous on X. Then by the Dini's theorem (see [17]), $\delta_N(\cdot, z) \to 0$ uniformly on a compact set X, as $N \to \infty$. ■

Proof of Corollary 2

Proof. Lemma 8 implies that there exists $\delta_{N_l}(x, z)$ such that

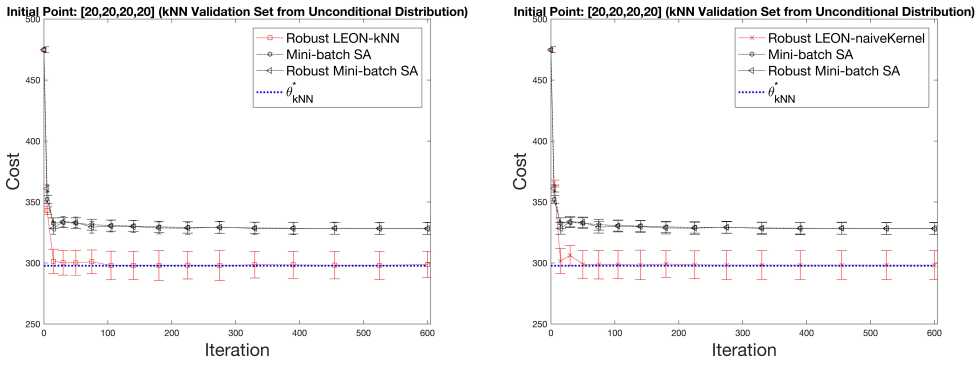
$$|\mathbb{E}[\hat{f}_{k,N_l}(x, z)] - f(x, z)| \leq \delta_{N_l}(x, z)$$

Since $\{N_l\}_l$ is an increasing sequence and $N_l \to \infty$ as $l \to \infty$, Lemma 8 also implies that $\delta_{N_l}(\cdot, z) \to 0$ uniformly on X, as $l \to \infty$. Lemma 8 also implies that $\delta_N(\cdot, z)$ is also monotonically decreasing with respect to N. Thus, all the conditions in Theorem 1 are satisfied, which completes the proof. ■

Evaluation of solution quality by *validation approach II*

Figures 4 and 5 show the computational performance of each algorithm in terms of solution quality based on *Validation Approach II*. It is worth noting that the interval shown in the figures are based on the

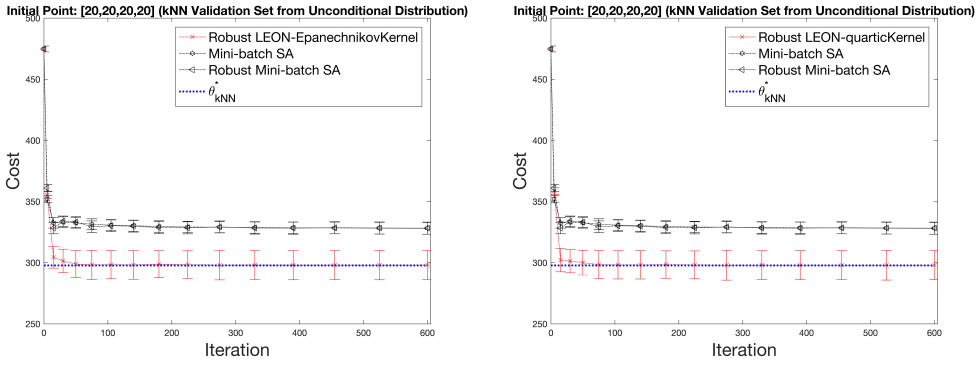
Figure 4: Computations of Robust LEON-kNN and Robust LEON-naiveKernel (*Validation Approach II*)



Initial point is $[20, 20, 20, 20]$, initial stepsize is 0.8.

deviation of kNN point estimate, which may not be a rigorous confidence interval. In particular, Figures 4 and 5 show that (Robust) Mini-batch SA converges to a different point, compared to the curves of Robust LEON.

Figure 5: Computations of Robust LEON-EpanechnikovKernel and Robust LEON-quarticKernel (*Validation Approach II*)



Initial point is $[20, 20, 20, 20]$, initial stepsize is 0.8.

Second initial setup in the numerical experiment

To measure the robustness of each approach, another setup in which the initial point is $[0, 0, 0, 0]$ and initial step size is 1 is used. The other parameters remain the same. The dataset generated in each iteration from the former experiment with the old setup (in which the initial point is $[20, 20, 20, 20]$ and initial stepsize is 0.8) is reused in this new experiment. One can identify from Table 2 that Robust LEON outperforms (Robust) Mini-batch SA in the second setup.

Table 2: Computational results of the second setup

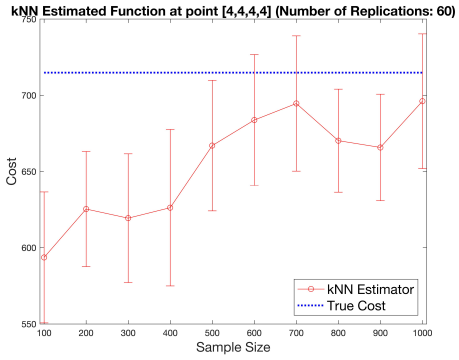
Algorithm	L	Obj(I)	Obj(II)
Mini-batch SA	600	1377.9($\pm$0.2400)	1376.8($\pm$2.3500)
Robust Mini-batch SA	600	327.8050($\pm$0.4280)	328.2610($\pm$4.9530)
Robust LEON-kNN	600	296.5760($\pm$0.9150)	298.8310($\pm$10.8080)
Robust LEON-naiveKernel	600	296.1550($\pm$1.0230)	298.2520($\pm$11.8990)
Robust LEON-EpanechnikovKernel	600	295.9950($\pm$1.0200)	298.0750($\pm$11.8620)
Robust LEON-quarticKernel	600	295.9280($\pm$1.0200)	297.9960($\pm$11.8610)

[a.] Observed predictor is $[-0.3626, 0.5871, -0.2987]$. Initial point is $[0, 0, 0, 0]$. Initial step-size is 1. Other setups are same as the ones in Table 1.

Bias of kNN Estimation

The same two-stage shipment planning problem is also used to numerically analyze the bias of kNN estimator, where we replicate the experiments 60 times. We estimate the values of $\hat{f}_{k,N}(x, z)$ (in (29)) and $f(x, z)$ (in (1)) at $x = [4, 4, 4, 4]$ and $z = [-0.3626, 0.5871, -0.2987]$. In particular, $\hat{f}_{k,N}(x, z)$ is evaluated by fixing $k = 20$ and increasing sample size, N. $f(x, z)$ is estimated by the average cost of one hundred thousand scenarios generated from the true conditional distribution given $Z = z$. Figure 6 indicates that the bias of kNN estimator $x = [4, 4, 4, 4]$ decreases as the sample size increases.

Figure 6: Bias analysis of kNN estimator



References

[1] N. S. Altman, *An introduction to kernel and nearest-neighbor nonparametric regression*, The American Statistician, 46 (1992), pp. 175–185.

[2] D. Bertsimas and N. Kallus, *From predictive to prescriptive analytics*, Management Science, (2019).

[3] R. Blundell and A. Duncan, *Kernel regression in empirical microeconomics*, Journal of Human Resources, (1998), pp. 62–87.

[4] P. E. Cheng, *Applications of kernel regression estimation: survey*, Communications in statistics-theory and methods, 19 (1990), pp. 4103–4134.

[5] E. Çinlar, *Probability and stochastics*, vol. 261, Springer Science & Business Media, 2011.

[6] L. Devroye, *The uniform convergence of nearest neighbor regression function estimators and their application in optimization*, IEEE Transactions on Information Theory, 24 (1978), pp. 142–151.

[7] L. Devroye, L. Gyorfi, A. Krzyzak, G. Lugosi, et al., *On the strong universal consistency of nearest neighbor regression function estimates*, The Annals of Statistics, 22 (1994), pp. 1371–1385.

[8] R. Durrett, *Probability: theory and examples*, vol. 49, Cambridge university press, 2019.

[9] Y. Ermoliev, *Stochastic quasigradient methods and their application to system optimization*, Stochastics: An International Journal of Probability and Stochastic Processes, 9 (1983), pp. 1–36.

[10] Y. M. Ermoliev and A. A. Gaivoronski, *Stochastic quasigradient methods for optimization of discrete event systems*, Annals of Operations Research, 39 (1992), pp. 1–39.

[11] E. Fix and J. L. Hodges Jr, *Discriminatory analysis-nonparametric discrimination: consistency properties*, tech. rep., California Univ Berkeley, 1951.

[12] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*, vol. 1, Springer series in statistics New York, 2001.

[13] W. Greblicki and A. Krzyzak, *Asymptotic properties of kernel estimates of a regression function*, Journal of Statistical Planning and Inference, 4 (1980), pp. 81–90.

[14] W. Greblicki, A. Krzyzak, M. Pawlak, et al., *Distribution-free pointwise consistency of kernel regression estimate*, The annals of Statistics, 12 (1984), pp. 1570–1575.

[15] L. Györfi, M. Kohler, A. Krzyzak, and H. Walk, *A distribution-free theory of nonparametric regression*, Springer Science & Business Media, 2002.

[16] G. A. Hanasusanto and D. Kuhn, *Robust data-driven dynamic programming*, in Advances in Neural Information Processing Systems, 2013, pp. 827–835.

[17] J. Jost, *Postmodern analysis*, Springer Science & Business Media, 2006.

[18] A. J. Kleywegt, A. Shapiro, and T. Homem-de Mello, *The sample average approximation method for stochastic discrete optimization*, SIAM Journal on Optimization, 12 (2002), pp. 479–502.

[19] A. S. Kozek, J. R. Leslie, E. F. Schuster, et al., *On a universal strong law of large numbers for conditional expectations*, Bernoulli, 4 (1998), pp. 143–165.

[20] J. Liu, G. Li, and S. Sen, *Coupled learning enabled stochastic programming with endogenous uncertainty*, Optimization Online, (2019).

[21] E. A. Nadaraya, *On estimating regression*, Theory of Probability & Its Applications, 9 (1964), pp. 141–142.

[22] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, *Robust stochastic approximation approach to stochastic programming*, SIAM Journal on optimization, 19 (2009), pp. 1574–1609.

[23] G. C. Pflug and A. Pichler, *From empirical observations to tree models for stochastic optimization: convergence properties*, SIAM Journal on Optimization, 26 (2016), pp. 1715–1740.

[24] B. T. Polyak and A. B. Juditsky, *Acceleration of stochastic approximation by averaging*, SIAM Journal on Control and Optimization, 30 (1992), pp. 838–855.

[25] H. Rahimian and S. Mehrotra, *Distributionally robust optimization: A review*, arXiv preprint arXiv:1908.05659, (2019).

[26] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on stochastic programming: modeling and theory*, SIAM, 2009.

[27] H. Takeda, S. Farsiu, and P. Milanfar, *Kernel regression for image processing and reconstruction*, IEEE Transactions on image processing, 16 (2007), pp. 349–366.

[28] H. Walk, *A universal strong law of large numbers for conditional expectations via nearest neighbors*, Journal of Multivariate Analysis, 99 (2008), pp. 1035–1050.

[29] ——, *Strong laws of large numbers and nonparametric estimation*, in Recent Developments in Applied Probability and Statistics, Springer, 2010, pp. 183–214.

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