Tackling Decision Dependency in Contextual Stochastic Optimization

Abstract: In this paper, we address the contextual stochastic optimization (CSO) problem, where decisions are made under uncertain parameters whose distributions can be inferred from covariates observed prior to decision-making. In practice, the distribution of uncertain parameters often depends not only on the covariates but also on the decisions themselves. For instance, in product pricing, the uncertain demand is influenced by the price set for the product. This phenomenon, referred to as the decision-dependent context, presents significant challenges in optimization.

Among the limited literature addressing this issue, Bertsimas and Kallus (2019) proposed a weighted sample average approximation (SAA) approach, where each sample in the SAA framework is weighted by a function of both covariates and decisions. However, since both the weights and the original loss function depend on the decision, the problem becomes computationally challenging to solve. In contrast, we directly apply the idea of weighted SAA to the gradient of the loss function, formulating what we term the contextual gradient. We integrate this contextual gradient into a gradient descent (CGD) algorithm to solve the CSO problem with decision-dependent effects. We prove that the proposed CGD method achieves a bounded error relative to global optimality under the strong convexity condition and converges to a stationary point of the expected gradient otherwise. Through extensive numerical experiments on real-world datasets, we demonstrate the superior performance of our proposed CGD algorithm compared to existing methods applicable to contextual optimization problems with decision-dependent effects.

Key words: contextual optimization, prescriptive analytics, decision-dependency

1 Introduction

In many real-world management processes, the managers regularly confront the decision subjected to the uncertainty. Such a problem is usually formulated into the stochastic programming as follows.

$$\min_{x\in\mathcal{X}} g(x) = \mathbb{E}_{y\sim f(y)}[l(x,y)], \tag{1}$$

where $l(x,y)$ is the objective function to be minimized, and x is the decision variable in a feasible region $\mathcal{X} \in \mathbb{R}^d$; y is some stochastic model parameters, whose probability density function is $f(y)$. In most situations, $f(y)$ is not known to the decision maker. Therefore, there are two complex tasks when solving this stochastic optimization problem: prediction and optimization. The prediction task requires the decision maker to predict stochastic parameters in models while the optimization task usually involves optimizing decisions with the aim of maximizing profits or minimizing costs based on the prediction result. When some side information z is provided to estimate y, *i.e.*, $f(y;z)$ is dependent on z, the problem is also called Contextual Stochastic Programming (CSO). We refer to Sadana et al. (2023) for a comprehensive review of

recent contextual optimization works. In the CSO context, problem (1) is implemented with a conditional expection on z, i.e.

$$\min_{x\in\mathcal{X}} g(x,z) = \mathbb{E}_{y\sim f(y|z)}[l(x,y)], \tag{2}$$

The CSO problem is widely applied in practice. A notable example is the contextual newsvendor problem (Ban and Rudin 2018). In this problem, the decision maker should optimize the order quantity x with unknown demand y. The objective is to minimize the negative of revenue $l(x,y) = -p(y \wedge x) + cx - s(x-y)^+$, where p, c, s are the per-unit price, inventory cost and salvage value respectively. And $\wedge$ denotes component-wise minimum, $(\cdot)^+ = \max\{\cdot, 0\}$. As discussed, the distribution of y is unknown but some features z, such as weathers, are available to the decision maker to the decision maker.

In the contextual newsvendor problem, the stochastic parameter (demand) is independent to the decision made (order quantity). However, in other applications, an additional challenge is that the stochastic parameter depends on decision variables. For instance, customer demand is usually affected by the pricing decision made in a revenue management context. Most of the existing solution approaches for CSO are not directly applicable to CSO with decision-dependent uncertainty. Most traditional approaches estimate the distribution of stochastic parameters conditioned on contextual information $y|z$, and the estimated conditional distribution is fixed in the downstream optimization step. In contrast, when the distributions of uncertain parameters is influenced by the decision, the conditional distribution $y|z,x$ changes continuously during the optimization step because of the iteration of x. Hence, it is hard to perform optimization unless we have the estimation of y conditioned on every possible combination of z and x. There are only a few researches focused on the decision dependent issue in CSO.

One way to be able to consider the decision effect in CSO is the weighted SAA approach proposed by (Bertsimas and Kallus 2019). Bertsimas and Kallus (2019) built a sample average approximation (SAA) model to estimate the expectation of objective function conditioned on x and z in (1). In their framework, a weight is assigned to each sample in the classic SAA as a function of x and z and the weighted sum of loss function over all samples is maximized. Although this approach achieves asymptotic optimality (see Theorem 10 in Bertsimas and Kallus (2019)), its downstream optimization task is difficult to solve. This is because the decision variables appear in both the estimation model and objective function, causing the non-convexity of the optimization model. Furthermore, for some discrete estimation models (e.g., kNN, tree models, and random forest), the approximate objective function is discontinuous to the decision variables even though $l(x,y)$ is continuous, which makes the ordinary gradient based algorithm inapplicable to solve such a model. The authors utilized discretization to keep the tractability and developed a special solution approach applicable for the tree-weight case. However, it remains unclear how to efficiently solve the general contextual optimization problem (1) under the decision-dependent effect.

The aim of this research is to provide a general approach to solve the decision-dependent contextual optimization models under the weighted SAA framework. We first apply the weighted SAA approach to

the gradient of the loss function, which we defined as *contextual gradient*. Compared to the gradient of the objective of the original weighted SAA, the contexual gradient does not require taking the derivative of the weights, which are obtained from complex machine learning methods and hard to get their derivative even if the derivatives exist. Under mild conditions, we show that the contextual gradient is an unbiased estimation of the expected derivative of the objective function and that the optimal solution should have zero expected derivative of the objective function. According to the properties, we embed the contextual gradient can to develop a contextual gradient descent (GD) to solve the decision-dependent CSO problem and establish a global convergence guarantee of the contextual gradient descent (CGD) algorithm when the objective function is convex. Specifically, we prove the error upper bounds of CGD under convex and strong convex conditions, which is separable into decision-dependent and initial value errors. Furthermore, under the strongly-convex condition, the CGD algorithm achieves a converging upper bound of the distance between the iterative solutions and the global optimal solution. We also extend the convergence results to the general non-convex case. Under the non-convex condition, although we cannot completely characterize global optimality guarantees, we find that the algorithm can converge to the stationary point of the original prescriptive optimization problem. We conduct extensive numerical experiments on both synthetic and real life data to demonstrate the effectiveness of the proposed CGD algorithm. The results show the outperformance of the proposed methods from both effectiveness and efficiency aspects.

1.1 Contributions

Our contributions are mainly three-fold:

First, we develop an effective metric of the optimality of the decision-dependent CSO problem. We propose the concept of contextual gradient. We prove that the contextual gradient is an unbiased estimation of the expected gradient of the objective function $l(x,y)$ (Proposition 1) and that the optimal solution should have a zero excepted gradient of the objective function. As such, the contextual gradient thereby provides a criterion of the optimality of the CSO problem and enables the design of gradient-based algorithms to solve the CSO problem with decision dependence.

Second, we provide an efficient and effective approach to solving the CSO problem with decision-dependent effect. We develop the contextual gradient descent (CGD) algorithm by embedding the contextual gradient into the gradient descent algorithm. We provide a thorough analysis on the convergence algorithm. Under mild assumption, we prove the convergence of the algorithm to the optimal solution and stationary point with and without the strong convex assumption, respectively.

Third, we show the practical values of our proposed CGD algorithm through extensive numerical results. Our numerical results show the superior performance of the proposed CGD algorithm. We conduct numerical experiments based on data that comes from both the simulation and practice. Compared to the

solving approach given by Bertsimas and Kallus (2019), our algorithm obtains a smaller gap in a significantly shorter time. Moreover, compared to the parametric estimate-then-optimize (ETO) solution approach that takes linear decision assumption to the stochastic parameter, our algorithm is more generalizable by its distribution-free and nonparametric setting, while the parametric ETO method need distributional assumptions on the decision dependency. Our algorithm outperforms by 33% under complex demand distribution. Even when the true demand is exactly a linear regression model, our algorithm shows $< 5\%$ gap to the parametric PTO approach that owns a correct prior knowledge to the problem.

1.2 Applications

Price-setting newsvendor. Although the application of end-to-end model to newsvendor problems is widely studied (Ban and Rudin 2018, Lin et al. 2022), the end-to-end price-setting newsvendor problem is seldom studied except for some options like quantile regression (Harsha et al. 2021). In the newsvendor pricing problem, the decision maker should also make decision on pricing, while the stochastic demand is dependent on the pricing decision, hence shows the decision-dependent property. In this case, the PTO framework does not work since we cannot optimize the pricing decision with fixed demand. Our framework can directly optimize the pricing decision based on the historical pricing and demand data.

Dynamic Facility Location. Mobile retail stores represent an innovative business model, where small, movable shops are deployed to different locations within a city. For instance, a coffee chain might use trucks to sell beverages in high-traffic areas. The demand for these mobile stores is highly influenced by their locations, as proximity and accessibility directly affect customer convenience. Moreover, demand fluctuates based on factors such as weather conditions, weekdays versus weekends, and local events, all of which are known in advance. This situation can be modeled as a decision-dependent CSO problem, where the decision is to dynamically adjust the set of store locations based on relevant covariates. When considering decisions on a continuous map, as in Cao et al. (2021), our framework can effectively solve this problem by utilizing historical location data, covariates, and demand data.

Project Portfolio Investment Allocation. Consider a company with a limited budget, evaluating how to allocate funds across a portfolio of projects. The company can leverage historical data and predictive analytics to estimate the potential returns based on covariates such as project features, market trends, and macroeconomic indicators. Furthermore, the return on investment is decision-dependent—the outcome of each project is influenced by the amount of investment allocated to it. For example, overfunding competing projects could lead to internal competition, diminishing the overall returns. Our method is well-suited to this scenario, helping determine the optimal investment allocation for each project.

1.3 Literature Review

CSO problem has been a focal topic in the operations research community in recent years. In this section, we provide a review on contextual optimization and stochastic models with decision dependency and compare them with our approach.

The estimate-then-optimize (ETO) is one of the most widely applied paradigms to solve CSO problems. ETO first estimates the conditional distributions of the random parameters given the context feature and then optimizes the decision under the conditional distribution. Our research falls into this stream of literature. The most related work is that of Bertsimas and Kallus (2019), which proposed the weighted sample average approximation (SAA) approach to solve the CSO optimization problem. In the approach, the conditional distributions were approximated by the weighted empirical distributions, where the weights represent the similarity between the historical context and the focal one and is obtained through ML methods, such as nearest neighbor and decision tree. Bertsimas and McCord (2019) further extended the work by adding the penalized term into the objective. Srivastava et al. (2021) further designed a regularized approximation to guarantee the out-of-sample performance based on the weights determined by the Nadaraya-Watson kernel regression. Lin et al. (2022) adopted this weighted SAA framework to a risk-averse newsvendor problem. However, most of these works assumed that the stochastic parameters are independent of the decision variable. Although the framework can take the decision-dependent effect by taking the decisions as inputs of the ML methods, they bypassed the computational difficulty brought by the effect. When the decisions are continuous, Bertsimas and Kallus (2019) only gave the solution approach when tree regression method is adopted in the estimation step. Otherwise, the decision-dependent model can only be solved by discretizating the continus variable, which significantly improve the computational difficulties.

Except for the weighted SAA, the residual-based approach is another ETO approach to solving the CSO problem. This approach assumes a regression type of uncertainty. That is, the mean of the uncertain parameter relies on the covariates while there is a noise independent of the covariates. ML methods, such as linear regression, decision tree (Ban et al. 2019, Kannan et al. 2022), are applied to learn the relationship between the uncertainty parameters and the covariates and the prediction residual is applied in the optimization. Another popular ETO approach is the smart predict-then-optimize framework proposed by Elmachtoub and Grigas (2022), which use the decision loss to calibrate the estimation model of the uncertainty parameter. Although the decision-dependent effect can be considered by taking the decision as part of input when training the estimation model of the uncertainty parameter, the tractability of the resulting optimization is not guaranteed.

Except for the ETO approaches, the CSO problem is also solved in an end-to-end way. That is, data is directly imported to the optimization model used to make decision. Some literature learned a parameterized policy as a function of covariates that directly optimizes the objective function optimization. For example,

Ban and Rudin (2018) adopted the linear decision rule and solved the newsvendor model by optimizing the parameters in the linear decision. They proved its superiority to SAA approach. Apart from the linear decision, Zhang and Gao (2017), Cristian et al. (2022), Oroojlooyjadid et al. (2020) adopted the neural network to parameterize the decision. Kallus and Mao (2022) applied a random forest as a proxy of the decision policy. However, these approaches still impose the independence of stochastic parameters on decisions. In contrast, our framework can estimate the expectation conditioned on both side information and decisions, and thus provide a clear way to deal with the decision-dependent property.

Recently, Feng and Shanthikumar (2023) proposed a more general framework named operational data analysis (ODA), which includes both ETO and end-to-end frameworks. The authors recognized that the data-driven decision under uncertainty is eventually a function or a statistic of the data. Based on the fact, the framework applied a data integration model to project the data to decision. Through the projection, the framework may exploit some desired property inherent in the decision-making problem. They show that for some problems, solutions of many commonly applied approaches fitted into the ODA framework. Without considering the decision dependent effect, the weighted methods with weights determined by the kernel smoothing share the same solution with ODA for a quality design problem. Although the idea behind ODA is general, the explicit form of the data integration heavily relies on that of the decision-making problem and is hardly to find for general problems. For example, in a following work Chu et al. (2024) that applied ODA to a price-setting newsvendor problem, a typical problem with decision-dependent uncertainty, the data integration model is obtained base on the linear or log-linear assumption on the relationship between price and demand. In contrast, our research is based on non-parametric methods, which is less influenced by the possible mis-specification of the decision-dependency.

To summary, most CSO frameworks assumed the independence of stochastic parameters on decision variables. It is hard to extend existing CSO to the decision-dependent context since the estimation of distribution conditioned on decision variables is seldom discussed. There are some works related to CSO problems with decision-dependent uncertainty (Bertsimas and Kallus 2019, Harsha et al. 2021), but they either focused on a specific problem (*e.g.* newsvendor pricing problem), or did not provide a way for optimization. To fix this gap, our framework provides a computable optimization approach by the proposed CGD algorithm.

Though the decision-dependent property has not been fully studied in the field of contextual optimization, it has been widely studied for stochastic programming models without considering the contextual information. Dupačová (2006) discussed the computational difficulties of decision-dependent stochastic programming and argued that the tractability of decision-dependent stochastic programming relies on the specific structure of the problem. Recent works focused on providing solutions that are free of specific problems. For example, Mendler-Dünner et al. (2020a) proposed a repeated gradient descent and provided an iterative approach to solve the decision-dependent SP problem. Liu et al. (2021) proposed a coupled learning-enabled optimization (CLEO) algorithm to solve the stochastic programming models

with decision-dependent uncertainty, which estimated the conditional distribution by local linear regression within a delicately designed trust region. Except for not considering the contextual information, the above approaches also differed from our work from the aspect of knowledge on the distribution of uncertainty parameters. Both Mendler-Dünner et al. (2020a) and Liu et al. (2021) assume that the samples can be obtained from the oracle distribution during the implementation of their algorithm. Whereas, our work is trained in an offline setting where the dataset has already been collected and no samples under new decisions can be obtained as a feedback to adjust the solutions dynamically.

The decision-dependent property is also studied in the robust optimization (RO) and distributionally robust optimization (DRO). Luo and Mehrotra (2020) constructed a decision-dependent robust ambiguity set with finite and known support of the uncertain parameter, and presented tractable reformulation of the commonly applied uncertainty set. Noyan et al. (2021) constructed the ambiguity set centered at a parametric decision dependent distribution and include all distribution closed enough with distance measured by Wasserstein distance. The relationship between the uncertainty set and decision is assumed to be known perfectly in these research. In contrast, our framework does not rely on the knowledge about the distribution family and the dependency form.

To summarize, most existing decision-dependent models adopted a parametric way to estimate the conditional distribution or assumed the regression relationship between decision variables and stochastic parameters. In contrast, we consider a nonparametric and distribution-free approach and utilize the context information to solve the problem. Hence, our algorithm can be used for more complex distribution and decision-dependency cases.

Organizations

The rest of the paper is organized as follows. In Section 2, we formally state the model and assumptions, as well as the intuition of the contextual gradient, we further investigate the convergence property of the contextual gradient. In Section 3, we propose the CGD algorithm and provide its convergence under convex and non-convex cases. In Section 4, we conduct the numerical experiment on the CGD algorithm and compare it with other existing solution approaches. Finally, Section 6 concludes the paper.

Preliminaries

For simplicity of notation, we use $x \wedge y$ to denote $\min\{x,y\}$, and $(x)^+$ to denote $\max\{x,0\}$. We let ∇ be the gradient denotation, and ∂ denote the subgradient set. Let $x = (x_1,...,x_d)^T$. The subscript denotes the corresponding coordinate of a vector, and $\mathbf{1}$ denotes the all-one vector. We use $\|\cdot\|$ to denote l_2 norm for vectors and matrix. A function f is $L-$Lipschitz continuous on $x \in \mathcal{X}$ if $\|f(x_1) - f(x_2)\| \leq L\|x_1 - x_2\|$ for any $x_1, x_2 \in \mathcal{X}$. A function f is $\gamma-$strongly convex if $f(x_1) - f(x_2) - \nabla f(x_1)^T(x_1 - x_2) \geq \gamma\|x_1 - x_2\|^2$. We use $[N] := \{1,...,N\}$ to denote the subscript set.

2 Problem Setting and Contextual Gradient Formulation

In this section, we formally introduce the decision-dependent contextual optimization problem, as well as the concept of the contextual gradient.

2.1 Problem Setting and Assumptions

Throughout the paper, we consider a contextual optimization model with decision dependency, where the distribution of uncertain parameters depends on both the contextual features and decision variables. Specifically, we focus on the task of minimizing the expected objective function given the contextual features and history datasets:

$$\begin{aligned} &\min_x g(x,z) \triangleq \mathbb{E}_{y\sim f(y|x,z)}[l(x,y)] \\ &\text{s.t. } x \in X, \end{aligned} \tag{3}$$

Here, x is the decision variable, z is the observed contextual information, and y is the random model parameter. Moreover, $f(y|x,z)$ denotes the probability density function of y, indicating that the uncertainty of y is dependent on x and z. The objective function is denoted by $l(\cdot,\cdot)$.

We make several assumptions about the optimization problem (3).

ASSUMPTION 1 **(Same bounded range)**. *The value range of a random parameter y remains the same under any x, and the value range of y is bounded.*

In practice, Assumption 1 is easily satisfied since we can take the union of the feasible region of y over all $x \in \mathcal{X}$ and assign the probability outside of the distribution $y|x$ as 0. We denote the range and its volume as Ω and S_Ω, respectively.

ASSUMPTION 2 **(Differentiation–integration exchange)**. *We assume that $l(x,y)$ is differentiable in X, and there exists $L^1(y)$ function $h(y)$, $|h(y)| \geq n|l(x+\frac{1}{n},y) - l(x,y)|$ for all $x \in X$ and $y \in \Omega$.*

Assumption 2 is the assumption of Lebesgue Dominated Convergence Theorem, which implies that we can change the order of integration and differentiation when calculating the derivative of the integral of $l(x,y)$ (see Theorem 2.27 in B.Folland (1999)). That is,

$$\nabla_x \int l(x,y)f(y|x,z)dy = \int \nabla_x (l(x,y)f(y|x,z))dy,$$

which enables us to access the derivative of the objective function. This assumption is quite general. It is satisfied as long as $l(x,y)$ is Lipschitz continuous in x for every y.

It seems that we do not need assumption 2 as long as we have Assumption 4

Another typical example is the revenue in pricing problem where x and y are the price and demand and $l(x,y) = xy$. In this case, $|l(x+\frac{1}{n},y) - l(x,y)|$

In Assumption 3, we assume that the distance between the decision-dependent distributions under different decisions can be bounded by the distance between the two decisions.

ASSUMPTION 3 **(ε-sensitivity)**. *We assume that the distribution map $f(y;\cdot,z)$ is ε-sensitive to decision variable. That is, for all x_1, x_2,*

$$W_1(f(y;x_1,z), f(y;x_2,z)) \leq \varepsilon \|x_1 - x_2\|_2, \tag{4}$$

where W_1 denotes the earth mover's distance (Rubner et al. 2000).

Intuitively, Assumption 3 ensures that the difference between decision-dependent distributions is not too large under different decisions x_1, x_2. The assumption holds for a wide range of distributions. As illustrated by Perdomo et al. (2020), when x only influences its mean and variance, a Gaussian family satisfy the assumption. In addition, when the decision dependent effect is in a regression format, i.e., the decision only influence the mean of the joint distribution of y and z, and the mean is Lipschitz continuous in x, we also have Assumption 3 satisfied. When analyzing the gap between the approximate solution and the optimal solution, Assumption 3 allows us to convert the difference between expectations under different distributions into the distance between decision variables.

ASSUMPTION 4 **(Lipschitz continuous and bounded gradient)**. *The cost function $l(x,y)$ is smooth and Lipschitz continuous with a Lipschitz gradient. Furthermore, its gradient in x in bounded. That is, the following five conditions hold*

$$\begin{aligned}
&(a)|l(x_1,y) - l(x_2,y)| \leq L_1\|x_1 - x_2\|, \quad \forall x_1, x_2 \in \mathcal{X}, y \in \Omega,\\
&(b)|l(x,y_1) - l(x,y_2)| \leq L_2\|y_1 - y_2\|, \quad \forall x \in \mathcal{X}, y_1, y_2 \in \Omega,\\
&(c)\|\nabla_x l(x_1,y) - \nabla_x l(x_2,y)\| \leq L_1^c\|x_1 - x_2\|, \quad \forall x_1, x_2 \in \mathcal{X}, y \in \Omega,\\
&(d)\|\nabla_x l(x,y_1) - \nabla_x l(x,y_2)\| \leq L_1^c\|y_1 - y_2\|, \quad \forall x \in \mathcal{X}, y_1, y_2 \in \Omega,\\
&(e)\|\nabla_x l(x,y)\| \leq L_3^c, \quad \forall x \in \mathcal{X}, y \in \Omega.
\end{aligned}$$

We assume the Lipschitz property of the objective function so that we can guarantee the approximate error of the contextual gradient will cause a bounded error in our algorithm. When the value ranges of variables are bounded, many commonly used loss functions, such as linear, quadratic, hinge, and logistic loss, naturally satisfy this assumption.

ASSUMPTION 5 **(Lipschitz continuous density)**. *The probability density function of y has a Lipschitz gradient. That is,*

$$\|\nabla_x f(y;x_1,z) - \nabla_x f(y;x_2,z)\| \leq L_{3g}\|x_1 - x_2\|, \quad \forall x_1, x_2 \in \mathcal{X}, y \in \Omega.$$

We also assume that the distribution function of random parameters has a limited change rate when the decision x changes. The assumption hold for many commonly applied distributions. For example, when the Gaussian distribution with mean and variance being the Lipschitz continuity in x, Assumption 5.

Moreover, they are only required in some of the convergence results.

2.2 Weighted-SAA Estimation

The specific form of $f(x,y,z)$ is seldom known in practice. Instead, a set of historical data is available $\{z^i, x^i, y^i\}, \forall i \in [N]$ where N is the number of records. Now, we discuss the weighted-SAA approach to approximate the expected objective function in (3) (Bertsimas and Kallus 2019, Lin et al. 2022), which is formally shown as follows.

$$\min_x \hat{g}(x,z) \stackrel{\Delta}{=} \sum_{i=1}^{N} w^i(x,z) l(x,y^i), \tag{5}$$

where $w^i(x,z)$ is a weight function to pool the data with different contextual information and decisions to predict the current scenarios. The basic idea of determining the weight is that data with similar contextual information and decisions to the current scenario should indicates more information of the distribution of random parameters under the current scenario. Accordingly, the weight of data point i is larger if the current scenario is more similar to the historical one recorded in the data point. These weights are always derived by the ML methods in advance of solving (5). We present the several examples in the following. A more thorough discussion on the implementation of the ML methods can be found in Bertsimas and Kallus (2019), Lin et al. (2022).

EXAMPLE 1 (KNN WEIGHT). The weight function can be derived from the definition of kNN:

$$w^{\text{kNN},i}(x,z) = \frac{1}{k}\mathbb{I}\{(x^i,z^i) \text{ is a kNN of } (x,z)\}, \quad \forall i \in [N], \tag{6}$$

where $\mathbb{I}\{\cdot\}$ is the indicator function, $[N] = \{1,2,...,N\}$ denotes the index set, and x^i is a kNN of x if and only if $|\{j \in \{1,...,N\} \backslash i : \|x^j - x\| < \|x^i, x\|\}| < k$.

EXAMPLE 2 (KERNEL REGRESSION WEIGHT). We can use the kernel function that measures the distances in (x,z) to construct the weight function:

$$w^{\text{KR},i}(x,z) = \frac{K_h\left((x,z) - (x^i,z^i)\right)}{\sum_{j=1}^{n} K_h((x,z) - (x^j,z^j))}, \tag{7}$$

where $K_h : \mathbb{R}^{dim(z)+1} \to \mathbb{R}$ is the kernel function with bandwidth h. Common kernel functions include the uniform kernel, triangular kernel and Gaussian kernel. If not noted, the kernel functions below refer to the Gaussian kernel function:

$$K(z) = \frac{1}{\sqrt{2\pi}} \exp^{-\|z\|_2^2/2}. \tag{8}$$

EXAMPLE 3 (CART WEIGHT). The CART weight functions are given by:

$$w^{\text{CART},i}(x,z) = \frac{\mathbb{I}\{R(x,z) = R(x^i,z^i)\}}{|\{j : R(x^j,z^j) = R(x,z)\}|}, \tag{9}$$

where $R : \mathcal{X} \times Z \to \{1,...,r\}$ is the function that maps features to the r leaves on the CART. In the CART, a leaf is a collection of sample points that are classified to the same group.

EXAMPLE 4 (RANDOM FOREST WEIGHT). The random forest weight functions are given by:

$$w^{\mathrm{RF},i}(x,z) = \frac{1}{N_E}\sum_{e=1}^{N_E} w^{\mathrm{CART},i,e}(x,z), \tag{10}$$

where N_E is the number of estimators in the random forest, and $w^{\mathrm{CART},i,e}(x,z)$ is the CART weight of the eth estimator in the random forest.

The approximate model (5) shares similar properties across the different ways of obtaining the weight function. Due to its better convergence property, we only present the results with the weight determined by the kernel method in as an example in the following two sections for brevity and compare several of them numerically to show their similarity in Section 4.

However, when x is included in the model weight, the approximate model (5) is hard to solve. First, there are cross-product terms since the decision variable exists in both the weight $w^i(x,z)$ and objective $l(x,y)$, thus the model can be non-convex even when $l(x,y)$ is convex on x. Second, the approximate model can be non-smooth and discontinuous if the discrete ML estimation models are adopted (e.g., kNN and tree models), even when $l(x,y)$ is smooth. In this case, traditional optimization approaches cannot be directly used for optimizing the weighted-SAA problem, and new methods are required for the estimation with the decision dependency taken into account. In the following sections, we introduce the concept of contextual gradient and show how to use it to mitigate the computational issue mentioned above.

2.3 Contextual Gradient

We formally describe the concept of contextual gradient as a weighted SAA of the gradient of the loss function, which is formally defined in Definition 1.

DEFINITION 1 (CONTEXTUAL GRADIENT). Given a contextual variable z' and a decision x', the contextual gradient at x' given dataset $\{z^i, x^i, y^i\}$ is defined as

$$\hat{G}_N(x';z') = \sum_{i=1}^{N} w^{(i)}(x',z')\nabla_x l(x',y^i), \tag{11}$$

where $w^{(i)}(\cdot,\cdot)$ is the weight function calculated by historical data.

The weights in (11) are obtained in the same way as those in (5). Comparing to directly take the derivative of the weighted SAA objective (5), we do not take the derivative of the weights when defining the contextual gradient. This avoids many computational issue as we discussed, both due to the complexity of ML methods used to obtain the weights and the possible non-convexity caused by the products between the loss functions and the weights. In the following example, we show the difference in the computation of contextual gradient (11) and the gradient of (5)

EXAMPLE 5 (PRICE-SETTING NEWSVENDOR PROBLEM). Consider the repeated selling of a perishable product whose demand is random and dependent on the price. The cost of procuring the product is c per unit. In the newsvendor pricing problem with contextual information, the decision maker need to jointly make the pricing p and ordering decision q to maximize its profit. Let $x = (p,q)^T$ When the demand realization is d, the profit given p,q is

$$l(x,y) = l(p,q,y) = -p(y \wedge q) + cq - s(q-y)^+.$$

Since $l(p,q,y)$ is not smooth when $q = y$, we study its contextual subgradient instead.

$$\partial_{p,q} l(p,q,y) = \left\{ \begin{bmatrix} -(y \wedge q) \\ -(p-c) + (p-s)e \end{bmatrix} : e \in [\mathbb{I}\{q > y\}, \mathbb{I}\{q \geq y\}] \right\}. \tag{12}$$

The subgradient set only contains one element almost everywhere. Once the weight is determined by the machine learning model, we can use (12) to calculate the contextual gradient. However, the gradient of (5) is not easy to calculate, which is equivalent to

$$\partial_{p,q} g(x,z) = \sum_{i=1}^{N} w^i(x,z) \partial_{p,q} l(p,q,y) + \partial_{p,q} w^i(x,z) l(p,q,y).$$

The major difficulty in calculation is $\partial_{p,q} w^i(x,z)$. For the kernel methods, it is hard to compute due to the exponential terms; for the weights obtained by the kNN or CART, the gradients of the weights do not exist. Whereas, the computation of the contextual gradient does not require the derivatives of the weights.

Lemma 1 show the statistical meaning of the contextual gradient (11). Specifically, it is an unbias estimation of the conditional expectation of the gradient of the loss function.

LEMMA 1 **(Convergence of Contextual Gradient (Theorem EC.9 in Bertsimas and Kallus (2019)))**. *Suppose that the joint distribution of (x,z) is absolutely continuous in the dataset, and its density function is bounded away from 0 to $+\infty$ on the support of x,z, and is twice continuously differentiable. Then the following uniform convergence over the inputs to the weights (x',z'), for some $c_N \to \infty$, almost surely,*

$$\lim_{N\to\infty} \sup_{\|x'\|+\|z'\| \leq c_N} \left| \hat{G}_N(x';z') - \mathbb{E}[\nabla_x l(x,y) | x = x', z = z'] \right| = 0, \tag{13}$$

for weight functions that base on kernel method defined in (7).

For other weight function defined in Section 2.2, though we do not have clear theoretical convergence of the contextual gradient, we have observe them to perform well numerically. Readers can refer to Section 4.2 for detailed information.

It is worth note that $\mathbb{E}[\nabla_x l(x,y) | x = x', z = z']$ is not the true gradient of the objective function (14), which is defined as

$$G(x';z') = \nabla_x \mathbb{E}[l(x,y) | z = z', x = x']. \tag{14}$$

As such, it is unclear how the contextual gradient is related to the optimality of the problem. Theorem 1 shows that $\mathbb{E}_{f(y;x,z)}[\nabla_x l(x,y)]$ can also be an indicator of the optimal solutions.

The gradient in Theorem 1 is not the one in Lemma 1. Can we make them the same? Or present a relationship between the two gradients.

THEOREM 1 (**Necessary condition of optimality**). *If x^* is the optimal solution for a decision-dependent problem* (3) *with loss function as an L_1 Lipschitz function and satisfying Assumptions 2 and 3, then* $\|\mathbb{E}_{f(y;x^*,z)}[\nabla_x l(x^*,y)]\| \leq L_1\varepsilon$.

Theorem 1 builds a connection between the contextual gradient and optimality condition. It indicates that one necessary condition for optimality is that the norm of the expected gradient should not be too large. As the contextual gradient is an unbias estimation of $\mathbb{E}_{f(y;x,z)}[\nabla_x l(x,y)]$, we can use it to reveal the optimality of the solutions. As such, we can apply it in the gradient-based algorithms, such as gradient descent algorithm, stochastic gradient descent algorithm, momentum optimization, etc., as an alterative to the true gradient (14). As we will show, the corresponding algorithm enjoys similar converging properties as the one embedding the true gradient.

3 Contextual Gradient Descent Algorithm

In this section, we focus on the case of non-constrained condition (*i.e.*, $\mathcal{X} = \mathbb{R}^d$) and embed the contextual gradient into a gradient descent algorithm 1, which is referred to CGD in the rest of the paper, to show its effectiveness in optimization.

Algorithm 1 The Contextual Gradient Descent Algorithm

Input: initial solution x^0, contextual information z, dataset $\{x^i, z^i, y^i\}_{i=1}^N$.

Output: solution $\hat{x}^*$.

1: $r = 0$
2: **while** Stop criteria not satisfied **do**
3: Calculate contextual gradient $\hat{G}_N(x^r; z)$ by (11)
4: Select step size η^r
5: Let $x^{r+1} = x^r - \eta^r \hat{G}_N(x^r; z)$
6: $r = r + 1$
7: **end while**
8: $x^* = x^r$

The contextual gradient descent follows a similar descending paradigm to the gradient descent algorithm. As other gradient descent algorithm, the step size in the CGD algorithm must be suitably determined to

guarantee the convergence of the algorithm. We adopt the diminishing step size and the Armijo step size in our work. The detailed introduction and comparison between two choices of step size can be seen in Appendix EC.3.2.

If the contextual gradient plays the same role as the true gradient, we can expect the convergence of CGD algorithm to a stationary point under general cases to global optimality under convex case. The convergence result of the CGD algorithm under the convex case and its error bound $|g(x_N^k) - g(x^*)|$ of to the optimal solution is proved in Theorem 2. In the strong convex case, we show a stronger convergence of the distance $\|x_N^k - x^*\|$ in Theorem 3. Furthermore, we extend our results to the general non-convex setting and analyze the convergence property of the CGD algorithm under two types of step size choices. We demonstrate that the CGD algorithm also converges to some stationary point with bounded expected gradient in a rate of $O(\varepsilon^{-2})$ where ε is a constant defined in Assumption 3. Despite the gap between the expected gradient and the true gradient of objective (3), we demonstrate that converging to some stationary point with a bounded expected gradient is a necessary condition of optimality in Theorem 1.

3.1 Convergence under Convex Case

In this subsection, we focus on the convergence guarantee of the CGD algorithm when $l(x,y)$ is convex on x. We first give the error bound under general convex case in Theorem 2.

THEOREM 2 **(Error Bound in Convex Case)**. *Suppose that $l(x,y)$ is convex on x and Assumptions 2, 3, 4(b) and 4(e) are satisfied. Denote x_N^r as the r th iteration of the CGD algorithm based on a dataset of N samples. Then for any small $\zeta > 0$, there exists N_0, when the sample size $N > N_0$, after k iterations,*

$$\min_{0\le r\le k}\{\mathbb{E}_{f(y;x_N^r,z)}[l(x_N^r,y)] - \mathbb{E}_{f(y;x^*,z)}[l(x^*,y)]\} \le \frac{3}{2}\zeta + \frac{\|x_N^0 - x^*\|^2 + (L_3^c)^2\sum_{r=0}^k(\eta^r)^2}{2\sum_{r=0}^k\eta^r} + \frac{\varepsilon L_2\sum_{r=0}^k\eta^r\|x^* - x_N^r\|}{\sum_{r=0}^k\eta^r}.$$

Theorem 2 characterizes the minimum functional error between the generated sequence x_N^r and the optimal solution x^* after k iterations.

Do we need specify the setting of step size and way to determine the kernel in this setting?

There are three terms on the right hand side representing different sources of error.

The first term is $\frac{3}{2}\zeta$ term is related to the approximation error of the original weighted SAA (5) and determined by the sample size collected. The term can be arbitrary small as the sample size goes to infinity.

The second term is a common one in the gradient descent algorithm. The term will decrease to 0 when the step size satisfies the condition that $\sum_{r=0}^k\eta^r \to \infty$, $\frac{\sum_{r=0}^k(\eta^r)^2}{\sum_{r=0}^k\eta^r} \to 0$, which is exactly the case when diminishing step policy is adopted. When constant step size is adopted, $\frac{\sum_{r=0}^k(\eta)^2}{\sum_{r=0}^k\eta} = \eta$. Namely, when the step size is small enough, the second term will also decrease to 0.

The last term is the caused by the decision-dependent effect (see proof of Theorem 2). Note that ε is the sensitivity of the conditional joint distribution of Y,Z on $X = x$ to the change in x. If $\varepsilon = 0$, namely, the

Y, Z are independent of X, the term equals to 0. This is exactly the case of applying the gradient descent algorithm to the weighted SAA without decision dependent effect. When $\varepsilon > 0$, we also note that the decision-dependent error will decrease as the solution x^r gets closer to the optimal solution x^*.

Consequently, it is important to know how the solution sequence converges to the optimal point. In the next, we investigate the distance to the optimal solution under the strongly convex condition. Mendler-Dünner et al. (2020a) proves the convergence result when the conditional distribution $y|x$ is known in advance. Similar to their work, we prove by bridging the CGD solution and optimal solution by an intermediate stable point.

DEFINITION 2 (STABLE POINT). Under contextual information z, the stable point is the fix point of the following iteration principle:

$$x_{PS} = \arg\min_x \mathbb{E}_{f(y;x_{PS},z)}[l(x,y)]. \tag{15}$$

We then state the distance bound between the solution sequence of the CGD algorithm and the stable point x_{PS} in Proposition 1.

PROPOSITION 1. *(**Distance to stable points**) Suppose that Assumptions 2, 3, 4(c) and 4(d) are satisfied, $l(x,y)$ is γ-strongly convex in x, and at least one stable point x_{PS} exists. We denote $A = \gamma - \varepsilon L_1^c$ and $B = L_1^c\sqrt{1+\varepsilon^2}$. For the case $A > 0$, we take a constant step size η that satisfies*

$$0 < 2B^2\eta^2 - 2A\eta + 1 < 1.$$

Then for any small $\zeta > 0$, set

$$UB(\eta,\zeta) = \frac{\zeta}{2(A-\eta B^2)}(1+\sqrt{2}\eta B + \sqrt{-2B^2\eta^2 + 2A\eta + (1+\sqrt{2}\eta B)^2}) \tag{16}$$

there exists a sample size N_0 such that, for all $N > N_0$, we have:

$$\lim_{K\to\infty} \|x_N^k - x_{PS}\| < UB, \tag{17}$$

and when $\|x_N^k - x_{PS}\| > UB$, $\|x_{N+1}^k - x_{PS}\| < \|x_N^k - x_{PS}\|$.

Namely, when the distance between x_N and the stable point x_{PS} is still relatively large, Algorithm 3 will produce x_{N+1} that is closer to the stable point. Ultimately, the distance between x_N to x_{PS} will be less than or equal to UB. And notice that,

$$\lim_{\eta\to 0} UB(\eta,\zeta) = \frac{\zeta}{2A}, \tag{18}$$

when the step size η is small enough, the upper bound $UB \in O(\zeta)$. Furthermore, when the sample size is big enough, that the estimation error ζ is sufficiently small, then the series of x_N will converge to the stable point x_{PS} almost surely.

Then we can discuss the convergence rate for this algorithm to the stable point.

PROPOSITION 2. *(**Convergence rate**) When $\|x_N^k - x_{PS}\| > 2UB$, the distance between x_{PS} and x_N^k converge at an exponential rate. Denote $C = \sqrt{1-\eta A+\eta^2 B^2} < 1$, we have,*

$$\|x_N^K - x_{PS}\| \leq \max\{2UB(\eta,\zeta), C^k\|x_N^0 - x_{PS}\|\}. \tag{19}$$

Proposition 1 relies on the constant step size, which violates the requirement in Theorem 2. How can we explain it? What if $A < 0$? Note that ε here is not the sensitivity defined in Assumption 3. We should revise it later. Whether do N_0, K_0 jointly or separately control ζ, ε?

We now focus on the distance to the optimal solution. Lemma 2 shows the relationship between the stable point and the optimal solution.

LEMMA 2 **(Theorem 4.3 in (Mendler-Dünner et al. 2020a))**. *Suppose that $l(x,y)$ is L_y-Lipschitz in y and strongly convex, and that Assumption 3 is satisfied. Then, for every stable point x_{PS}, we have*

$$\|x^* - x_{PS}\| \leq \frac{2L_y\varepsilon}{\gamma}.$$

Combining Proposition 2 and Lemma 2, we can now state the upper bound of the solution error of CGD algorithm.

THEOREM 3 **(Error bound under strongly convex case)**. *Denote x^* as the optimal solution of $g(x,z) = \mathbb{E}_{f(y;x,z)}[l(x,y)]$ and $l(x,y)$ is $\gamma-$strongly convex. Set the step size η according to Proposition 1. After k iterations, for any $\zeta > 0$, there exists a sample size N_0 such that, if $N > N_0$, the solution gap is bounded by*

$$|x_N^k - x^*| \leq \frac{2L_2\varepsilon}{\gamma} + \max\{2UB(\eta,\zeta), C^k\|x_N^0 - x_{PS}\|\}.$$

Theorem 3 shows that the upper bound of the distance between the solution of CGD algorithm and the optimal one reduces as the number of sample and iterations increases. As these number goes to infinity, the upper bound converges to a constant proportionate to the sensitivity of the conditional joint distribution shift caused by the change in x and inversely to the extent of strong convexity. This indicates when the loss function is sensitive to the solution while the distribution is not that sensitive.

With Theorem 3 at hand, we can give an analysis on the third term of Theorem 2. Denote $K = \lceil \frac{log(2UB(\eta,\xi)) - log(\|x_N^0 - x_{PS}\|)}{logC} \rceil$, we have that when $k \geq K$, $\|x_N^k - x_{PS}\| \leq 2UB(\eta,\xi)$. So we have

$$\sum_{r=0}^{k}\|x_N^r - x_{PS}\| \leq \begin{cases} \dfrac{\|x_N^0 - x_{PS}\|(1-C^{k+1})}{1-C}, & k < K, \\ \dfrac{\|x_N^0 - x_{PS}\|(1-C^{K+1})}{1-C} + 2(k-K)UB(\eta,\xi), & o.w. \end{cases} \tag{20}$$

Hence

$$\begin{aligned}
&\frac{\varepsilon L_2 \sum_{r=0}^{k} \eta^r \|x^* - x_N^r\|}{\sum_{r=0}^{k} \eta^r} \\
&= \frac{\varepsilon L_2 \sum_{r=0}^{k} \eta \|x^* - x_N^r\|}{\sum_{r=0}^{k} \eta} \\
&= \frac{\varepsilon L_2}{k} \sum_{r=0}^{k} \|x^* - x_N^r\| \\
&\leq \begin{cases} \frac{\varepsilon L_2}{k} \frac{\|x_N^0 - x_{PS}\|(1-C^{k+1})}{1-C}, & k < K, \\ \frac{\varepsilon L_2}{k} \left(\frac{\|x_N^0 - x_{PS}\|(1-C^{K+1})}{1-C} + 2(k-K)UB(\eta,\xi) \right). & o.w. \end{cases}
\end{aligned} \tag{21}$$

And when k is sufficiently big, $\frac{\varepsilon L_2}{k} \frac{\|x_N^0 - x_{PS}\|(1-C^{K+1})}{1-C}$ *becomes zero,* $\lim_{k\to\infty} \frac{\varepsilon L_2}{k} \sum_{r=0}^{k} \|x^* - x_N^r\| = 2\varepsilon L_2 UB(\eta,\xi)$. *As we mentioned earlier, when the step size is chosen sufficiently small and the sample amount N is big enough that* ξ *is sufficiently small,* $UB(\eta,\xi)$ *decrease to zero. Thus the third term in theorem 2 also decreases to zero when the aforementioned conditions are satisfied.*

Does this mean that the gap of CGD is 0 in the ideal case?

REMARK 1. As discussed in Mendler-Dünner et al. (2020a), constant A is a sharp threshold for the convergence of gradient descent. When $A \leq 0$, the algorithm can fail to converge to the stable point since it might not exists. Of course, this does not mean that the bound given by Theorem 2 is invalid. Although we cannot give the upper limit of the third term in Theorem 2, the numerical experiment results show that a certain reduction rate can still be maintained. We discuss the situation where $A \leq 0$ or even l is non convex in Section 3.2.

3.2 Convergence under general setting

To investigate the performance of the CGD algorithm under more general cases, we extend the convergence analysis to this non-convex case. Similar to the convergence result of typical gradient descent, we study the convergence of CGD algorithm to a stationary point, that is, the point x^* where ~~$\mathbb{E}_{f(y;x^*,z)}[\nabla_x l(x^*,y)]$~~ is small.

A new definition is needed. $\mathbb{E}_{f(y;x^*,z)}[\nabla_x l(x^*,y)] = 0$ is not the definition of stationary point. However, this whole section is hoping to state that the true optima possesses small $\mathbb{E}_{f(y;x^*,z)}[\nabla_x l(x^*,y)]$ and the converged point has limited $\mathbb{E}_{f(y;x^*,z)}[\nabla_x l(x^*,y)]$.

The properties mentioned in this section indeed not only hold in non-convex case but also in convex case.

PROPOSITION 3 **(Convergence under diminishing step size)**. *Suppose Assumptions 2, 3, 1, 5 and 4(a)-(c) hold, that the objective function* $l(x,y)$ *is twice differentiable in x and its absolute value is bounded by a*

constant L_4. If the gradient of the distribution density is also bounded by a constant L_5, and the step size η^r is diminishing with $\sum_{r=0}^{\infty}\eta^r=\infty$, there exists a sample size N_0, when $N>N_0$, any limit point of the sequence generated by the CGD algorithm is a stationary point of the cost gradient expectation.

$$\text{if } \lim_{N\to\infty}\lim_{r(\in\mathcal{K})\to\infty} x_N^r=\bar{x},\ \text{then } \mathbb{E}_{f(y;\bar{x},z)}[\nabla_x l(\bar{x},y)]=0. \tag{22}$$

PROPOSITION 4 **(Convergence under Armijo step size)**. *Under Assumptions 2, 3, and 4(a), suppose that the CGD algorithm adopts the Armijo step size with σWhat's the meaning of σ?, and that the sample size N is sufficiently large. Then, any limit point $\bar{x}$ of the sequence generated by the CGD algorithm has a bounded expected gradient.*

$$\text{if } \lim_{N\to\infty}\lim_{r(\in\mathcal{K})\to\infty} x_N^r=\bar{x},\ \text{then } \|\mathbb{E}_{f(y;\bar{x},z)}[\nabla_x l(\bar{x},y)]\|\le\frac{\varepsilon L_1}{1-\sigma}. \tag{23}$$

What's the relationship between the stationary point of the expected gradient to the true problem?

Propositions 3 and 4 state that the CGD algorithm converges to the stationary point of the expected gradient. Note that this conclusion relies on Assumptions 1 and 5, which are strong conditions and may not be generally satisfied. Furthermore, when the constants S_Ω, L_4, L_5 are large, this convergence result may have a poor performance in practice. Compared with the diminishing step size, the expected gradient for CGD with Armijo step size is not guaranteed to converge to 0, but this convergence result holds under a milder condition where Assumptions 1 and 5 may not hold. And we can then derive from Theorem 1 that (23) is a necessary condition of optimality.

Then we investigate the convergence rate of the CGD algorithm. Compared with typical gradient descent, which only requires $O(1/\varepsilon^2)$ iterations to obtain an $\varepsilon-$stationary solution, the CGD algorithm also requires $O(1/\varepsilon^2)$ steps to converge to a range with expected gradient upper bound.

We have defined the ε-sensitive in Assumption 3. It does not mean the same here. Change the symbol here.

PROPOSITION 5. *Suppose Assumptions 2, 3, 4 and 4(c) hold, when $N\to\infty$, with fix step size $\eta\le\min\{\frac{1}{L_{1g}},\frac{1}{L_3^c}\}$, we have*

$$\min_{r=0,\dots,k}\|\mathbb{E}_{f(y;x_N^r,z)}[\nabla_x l(x_N^r,y)]\|^2\le\frac{2(\mathbb{E}_{f(y;x_N^0,z)}[l(x_N^0,y)]-\mathbb{E}_{f(y;x^*,z)}[l(x^*,y)])}{\eta(k+1)}+\frac{L_3^c L_1\varepsilon}{2}. \tag{24}$$

Like the convergence result in Proposition 4, Proposition 5 shows that CGD with constant step size will converge to a point with limited expected gradient, and the convergence rate is $O(\varepsilon^{-2})$, which corresponds to the $O(\varepsilon^{-2})$ lower bound of Agarwal et al. (2012). The bias term $\frac{L_3^c L_1\varepsilon}{2}$ to stationary point implies the error caused by decision dependency. In specific, it rises from the heterogeneous distribution under different decisions.

What's the significance of these propositions? The stationary point to the expected gradient is not the stationary point to the original problem. Can we use these proposition to illustrate the effectiveness of different method to set the step size?

4 Numerical Results

In this section, we validate the convergence performance of the CGD algorithm and compare its performance against other methods. First, we show the convergence of the CGD algorithm both in convex and non-convex cases with weights determined by different machine learning methods. Then, we show the Specifically, we compare the performance of the proposed CGD algorithm with the prescriptive approach in Bertsimas and Kallus (2019) and the estimate-then-optimize approach under the linear decision assumption. All computations were carried out in Python 3.10 on an Intel i7-9750H processor with 32.0 GB of RAM.

4.1 Data Description and Experiment Setup

We show the effectiveness of our proposed CGD algorithm through a price-setting newsvender problem in Example 5. The data generation setting is similar to Lin et al. (2022) where a newsvendor problem without setting the price is considered. The covariates are independently drawn from a 4-dimension Gaussian distribution $N(0,\Sigma)$, where Σ is a matrix $(1,0,0,0;0,2,0,0;0,0,3,0;0,0,0,4)$ and the price is drawn from (the scheme of generating the price).

I can not find how the price is generated in the training samples.

We focus on two demand models to generate the demand.

- **Complex Model:** The first follows the following scheme.

$$D = \max\{0, 60 - p + 12a^T(X + 0.25\phi) + 5b^T X\theta\}, \tag{25}$$

where $\phi \sim N(0, I_4)$ is a 4-dimensional vector, $\theta \sim N(0,1)$ is also a Guassian parameter. Both θ and ϕ are the stochastic factors that cause demand fluctuation. The constant vector $a = (0,8,1,1,1)^T$ and $b = (-1,1,0,0)^T$.

- **Linear Model:** The second demand model is simply a linear regression model to suit the PTO assumption. In this situation, we let

$$D = 60 - p + (1,1,1,1)^T z + \phi,$$

where $\phi \sim N(0,1)$. Thus, the demand follows a normal distribution under any price and feature.

Compare to the first model, the second one is more easily to fit. As such, it is used to show the effectiveness of our model when the benchmarkes are less misspeicified.

For our method, the hyperparameters of each ML method (e.g., kNN, kernel regression, CART, and RF weighting) are tuned through a grid search and the initial values of the CGD algorithm are $(p_0, q_0) =$

$(15, 30)$. The iteration stops when the step size is below 10^{-5} or the objective value is closed enough to the lower bound.

We evaluate the performance by the *optimality gap*, *i.e.*,

$$\text{gap} = (\mathbb{E}_{f(y;x^*,z)}[l(x^*, y)] - \mathbb{E}_{f(y;x,z)}[l(x, y)])/\mathbb{E}_{f(y;x^*,z)}[l(x^*, y)].$$

The optimal solution is easily to find as we know the true distribution. Note that while the true distribution $f(y|x, z)$ is known when generating the data and evaluating the results, it is unknown when we are solving the problem by CGD algorithm.

4.2 Convergence Performance

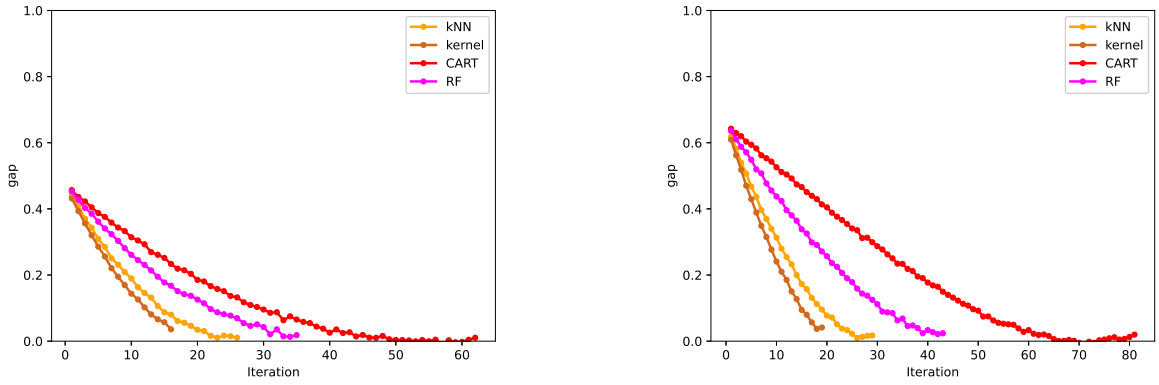
In this section, we validate the convergence performance of the CGD algorithm from different aspects.

4.2.1 Convergence under convex case.

We first perform the convergence of CGD algorithm under different weighting approaches under the convex price-only newsvendor pricing problem (*i.e.*, fixing the ordering decision). When adding a quadratic penalize, (specialize the term), term into the objective function $l(x, y)$, we extend the problem to a strongly convex case.

What value the ordering quantity is fixed to here?

The convergence results to these two cases are shown in Figure 1(a) and 1(b). The figures shows how the gaps of the CGD algorithms with different ML methods to determining the weights.



(a) Convergence under convex case

(b) Convergence under strongly-convex case

Figure 1 Comparison CGD algorithm with different weight under convex and non-convex conditions on D_{simu}.

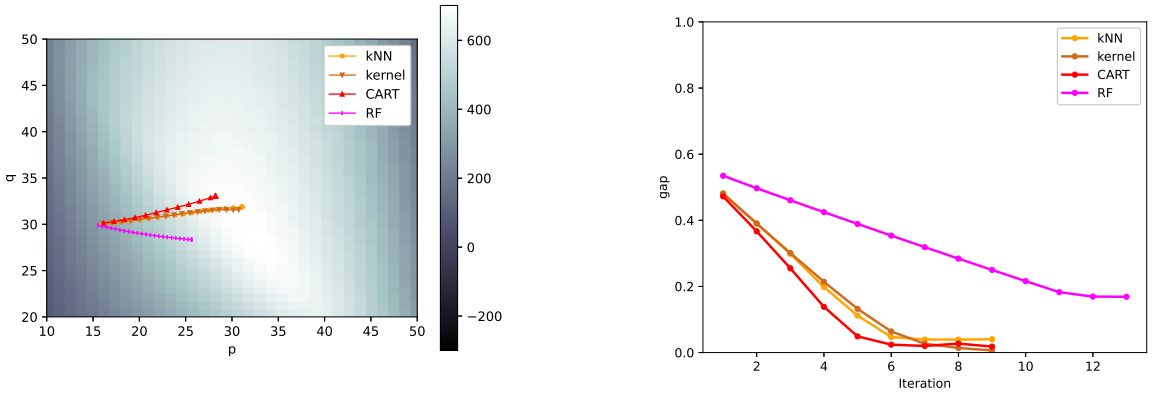
We observe that with different ML methods to determine the weights, the CGD algorithm converges well before the iterations stop. In both convex and strongly convex cases, the optimality gaps are less than 5%. This observation validates the convergence results in the strongly convex case discussed in Section **??** and indicates the effectiveness of the proposed method in the convex case.

4.2.2 Convergence under non-convex case.

We then investigate the convergence of the algorithm under the non-convex setting. Specifically, we jointly optimize the pricing and ordering decision in the newsvendor pricing problem, where $l(x,y)$ is non-convex to the decisions $x=(p,q)$. Figure 2(a) is a snapshot of the convergence process of the CGD algorithm with different ML method determining the weight. The figure shows that all four models descend toward the local optimum.

I cannot observe it. Maybe we can use a 3D figure to show that. Or we can just mark the local optimal solutions.

The optimality gaps to the local(global?) optimum are shown in Figure 2(b). As the figure shows, the CGD algorithm converge to a local optimum in a few iterations. Kernel regression and CART exhibit the best performance, indicating that they have more accurate estimates of profit and gradient. Among the four weight functions, the optimality gaps of kNN, kernel regression, and CART are less than 5%. Note that the priority of these two weighting methods does not always hold true, but depends on whether the weight method gives an accurate prescription to the conditional distribution of the demand.



(a) Process snapshot (b) Optimality gap

Figure 2 Convergence of CGD algorithm with different weight under non-convex condition on D_{simu}.

4.2.3 Sample Efficiency.

To verify the performance of the model under small sample conditions, we study the performance of the CGD algorithm under different sample sizes. We use the kNN weight method with $k=20$ and generate five stochastic instances for each sample size.

The number of repeats here is to little. And we may want to show the results of all methods.

The average, maximum, and minimum optimality gaps are shown in Figure 3. As the figure shows, the CGD algorithm performs well on average. The average gap is less than 10% over the repeats. Moreover,

the optimality gap becomes more stable and decreases as the sample size increases. This is because a larger sample size provides more information about the true distribution.

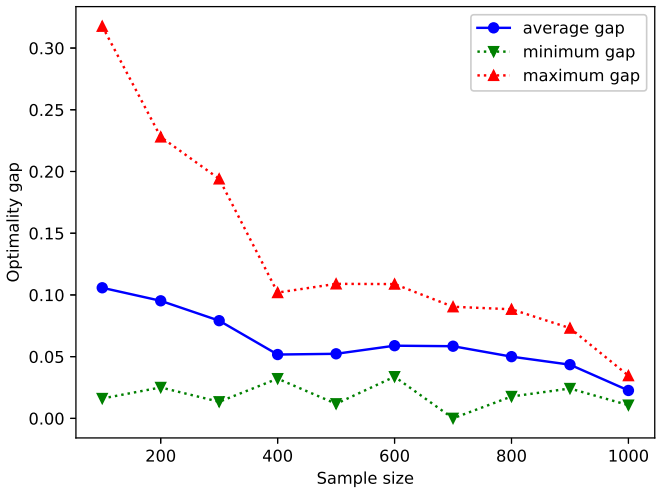


Figure 3 Optimality gap under different sample sizes on D_{simu}.

4.3 Comparison to Other Methods

4.3.1 Comparison with directly solving the weighted SAA model

Since we employ the weighted Sample Average Approximation (SAA) approach to develop the context gradient and the corresponding CGD algorithm, a natural question arises: how does the performance of the CGD algorithm compare to directly solving the weighted SAA problem? However, solving the weighted SAA problem with decision-dependent effects presents significant computational challenges. To demonstrate the effectiveness of the proposed method, we adopt two strategies for solving the problem:

- The first approach is the discretization technique introduced by Bertsimas and Kallus (2019), where decision variables are discretized to maintain the tractability of the model. This approach is applicable for weights determined by any machine learning method.
- The second approach involves applying the gradient descent algorithm directly to the objective function of the weighted SAA, as shown in equation (5). Note that this method is only applicable for weights that possess gradients, such as those derived from kernel regression.

We have conducted a comparison of both the optimality gap and computational time for the two models, with the results summarized in Table 1. When compared to the discretization method, the CGD algorithm demonstrates a comparable optimality gap in most instances and exhibits greater stability across various

Table 1 Optimality gap and running time comparison between CGD and PRE+DIS.

Strategies	kNN		kernel		CART		RF	
	gap	time (sec)	gap	time (sec)	gap	time (sec)	gap	time (sec)
Discretization	10.27%	60.37	0.94%	118.14	0.96%	14.11	114.44%	91.75
Gradient Descent	-	-			-	-	-	-
CGD	1.48%	1.92	0.56%	2.04	1.99%	1.07	4.97%	5.87

machine learning models. For instance, under random forest estimation, the optimality gap for the discretization strategy is 114.44%, whereas for the CGD algorithm, it is a significantly lower 4.97%. Furthermore, the CGD algorithm requires computational time that is ten times shorter than that of the discretization method.

In the case of the gradient descent algorithm, the CGD algorithm shows a marked improvement in performance when kernel regression is utilized to determine the weights. This observation aligns with our discussion in Example 5 that directly applying the gradient descent algorithm introduces additional nonconvexity, making it more prone to converging to suboptimal solutions. The findings above not only validate the effectiveness of our approach but also highlight the distinction of our proposed method from simply applying the gradient descent algorithm to the weighted SAA approach.

How many repeats do we implement to get the results? Complement the results of gradient descent here.

4.3.2 Comparison to other benchmarks

In this section, we compare the CGD model with several additional benchmarks.

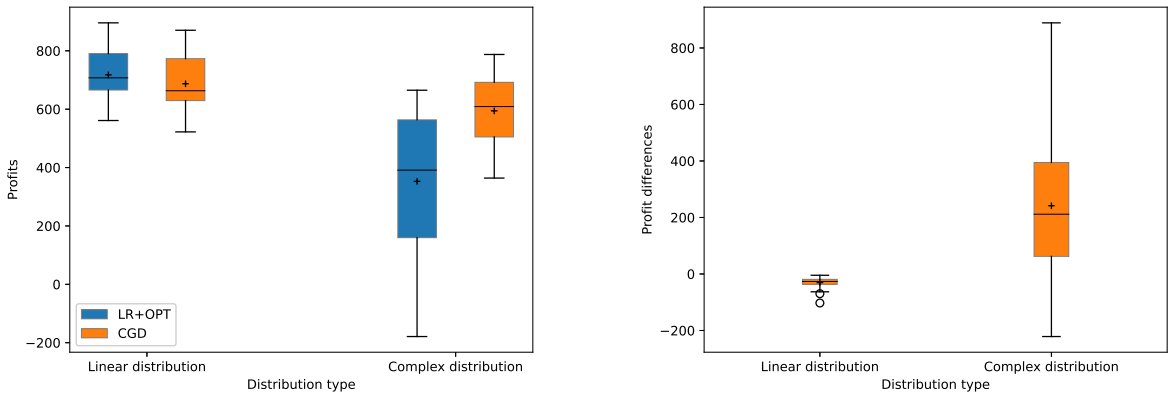
One benchmark is a widely used estimate-then-optimize framework in practice, which employs a linear decision rule. This framework first constructs a linear regression model of y with respect to the decision variable x and contextual information z, and then optimizes the decision based on the regression model (referred to as LR+OPT) (Ban and Rudin 2018, Demirović et al. 2019). Specifically, the LR+OPT framework begins by estimating a linear regression model:

$$\hat{y}(x,z) = \alpha_0 + \alpha^T(x,z),$$

where α_0 and α are the coefficients of the linear model. Subsequently, it replaces the model parameter y with $\hat{y}(x,z)$ and proceeds to optimize x directly. This approach provides a baseline for evaluating the performance of the CGD model in comparison to traditional methods.

The other benchmark is the ODA framework implemented by Chu et al. (2024) where the demand is assumed to be a linear function of the price.

Complement the description of the ODA framework and the results.



(a) Profits comparison between CGD and LR+OPT under two demand models

(b) Profit differences of CGD and LR+OPT under two demand models

Figure 4 Comparison between CGD and LR+OPT approaches under two demand models

The comparison is implemented against both the Complex and Linear demand model. The linear demand model complies with the assumption of linear regression and the ODA approach where the approach only suffers little mis-specification.

The results of our experiments are shown in Figure 4, where Figure 4(a) denotes the profit performance on the test dataset, and Figure 4(b) shows the difference of CGD profit minus the LR+OPT profit on the test dataset. We can observe that under the linear demand model, the LR+OPT performs slightly better than the CGD method. This result is not surprising because the true demand model satisfies the demand prediction assumptions exactly. Compared with the benchmarks framework, the CGD algorithm adopts a nonparametric approach to model the decision-dependent relationship, making it generalizable to complex distribution cases. In the complex demand distribution scenario, our CGD method significantly outperforms the LR+OPT framework. This result illustrates the generalization ability of the CGD algorithm.

The second figure is meaningless. We can delete it. I found that there are no descriptions on the sample size and number of repeats. Can you find it for me from the code?

In summary, when there is little information about the distribution of stochastic parameters, adopting the distribution-free method results in better adaptation to a range of real-world scenarios, leading to more robust solutions than assuming a specific distribution and decision-dependency rule for the stochastic parameters.

5 Case Study

We then test the performance of the CGD algorithm under the practical pricing problem in the electricity industry. The dataset comes from a real-world power plant pricing scenario[1]. This dataset describes the

[1] https://www.kaggle.com/datasets/aramacus/electricity-demand-in-victoria-australia

electricity demand and price situation in Victoria, Australia from 2015 to 2020. The descriptive information of the real data is shown in Table 2.

Table 2 Description of real electricity pricing data

Variable	Type	Description	Statistics
Date	datetime	the date of the recording	min 1Jan15, max 6Oct20
Demand	float	a total daily electricity demand in MWh	min 85.1k, median 120k, max 171k
RRP	float	a recommended retail price in AUD$/MWh	min 0, median 66.7, max 300
min temperature	float	minimum temperature during the day in Celsius	min 0.6, median 11.3, max 28
max temperature	float	maximum temperature during the day in Celsius	min 9, median 19.1, max 43.5
solar exposure	float	total daily sunlight energy in MJ/m$\hat{2}$	min 0.7, median 12.7, max 33.3
rainfall	float	daily rainfall in mm	min 0, median 0, max 54.6
school day	boolean	if students were at school on that day	True 69%, False 31%
holiday	boolean	if the day was a state or national holiday	True 4%, False 96%

The factors that influence daily demand are *price, temperature, solar exposure, school day* and *holiday*. We perform an artificial transformation on the temperature. We define heating degree day (HDD) as $HDD = (T_{min} - 18)^+$, and cooling heating degree day (CDD) as $CDD = (15 - T_{max})^+$, where T_{max} and T_{min} are the highest and lowest centigrade temperatures in one day. This transformation can better reflect the relationship between temperature and electricity demand. Note that the scales of features are different, so we standardize the feature to $[0, 1]$ when processing the data.

We employ a deep neural network (DNN) to model the relationship between demand and its influencing factors. The dataset is split into a training set ($n = 1895$, 90%) and a test set ($n = 211$, 10%). The training set is used to train the DNN, while its predictive performance is evaluated on the test set.

To implement the CGD algorithm, we bootstrap from the entire dataset to construct historical data. Additionally, we continuously collect new samples to assess the algorithm's performance. The demand values are predicted and labeled using the trained DNN. This approach ensures robust evaluation and adaptability of the CGD algorithm in dynamic scenarios. Figure 5 compare the profit collected from the CGD solution and historical price.

6 Conclusion and Future Directions

In this paper, we propose a novel approach to solve the contextual optimization problem under decision dependency. Compared with existing policies, the contextual gradient retains the first-order information of the objective function, and thus is more efficient than the discretization approach. Our CGD algorithm also

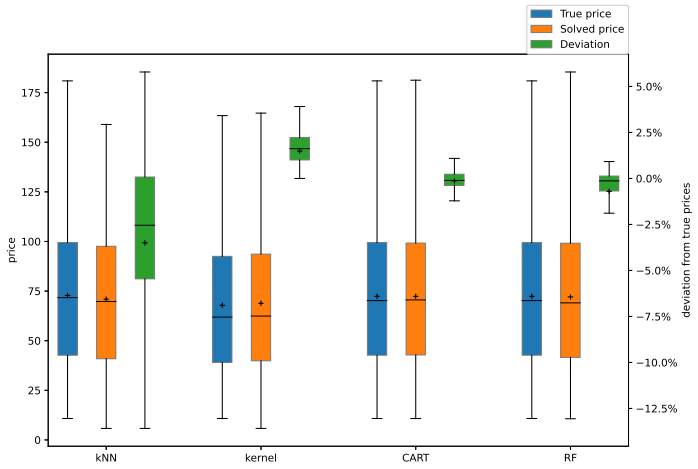


Figure 5 Comparison between real pricing decision and output pricing decision on D_{real}.

has a strong theoretical convergence guarantee under both the convex and non-convex cases and has a great generalization ability because the method is fully nonparametric.

Much remains open and requires further investigation. First, there may exist other algorithm designs based on contextual gradient. For example, one can embed the contextual gradient into the proximal gradient descent algorithm and stochastic gradient descent algorithm. One may characterize distance convergence properties. Second, In this paper, the convergence result is limited to the unconstrained setting. Efficient algorithms are still absent for solving the constraint contextual optimization problem under decision dependency.

References

Agarwal, Alekh, Peter L. Bartlett, Pradeep Ravikumar, Martin J. Wainwright. 2012. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *IEEE Transactions on Information Theory*, 58 (5), 3235-3249. doi:10.1109/TIT.2011.2182178.

Ban, Gah-Yi, Jérémie Gallien, Adam J Mersereau. 2019. Dynamic procurement of new products with covariate information: The residual tree method. *Manufacturing & Service Operations Management*, 21 (4), 798-815.

Ban, Gah-Yi, Cynthia Rudin. 2018. The big data newsvendor: Practical insights from machine learning. *Operations Research*, 67 (1), 90-108.

Bertsimas, Dimitris, Nathan Kallus. 2019. From predictive to prescriptive analytics. *Management Science*, 66 (3), 1025-1044.

Bertsimas, Dimitris, Christopher McCord. 2019. Optimization over continuous and multi-dimensional decisions with observational data. *Neural Information Processing Systems*.

B.Folland, Gerald. 1999. *Real Analysis Modern Techniques and Their Applications 2nd Edition*.

Cao, Junyu, Wei Qi, Yan Zhang. 2021. Online facility location. *Available at SSRN 3930617*, .

Chu, Leon Yang, Qi Feng, J George Shanthikumar, Zuo-Jun Max Shen, Jian Wu. 2024. Solving the price-setting newsvendor problem with parametric operational data analytics (oda). *Management Science*, .

Cristian, Rares, Pavithra Harsha, Georgia Perakis, Brian Quanz, Ioannis Spantidakis. 2022. End-to-end learning via constraint-enforcing approximators for linear programs with applications to supply chains. URL https://api.semanticscholar.org/CorpusID:259953167.

Demirović, Emir, Peter J Stuckey, James Bailey, Jeffrey Chan, Christopher Leckie, Kotagiri Ramamohanarao, Tias Guns. 2019. Predict+ optimise with ranking objectives: Exhaustively learning linear functions. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*. International Joint Conferences on Artificial Intelligence, 1078-1085.

Dupačová, Jitka. 2006. Optimization under exogenous and endogenous uncertainty. doi:10.13140/2.1.2682.2089.

Elmachtoub, Adam N., Paul Grigas. 2022. Smart "predict, then optimize". *Management Science*, 68 (1), 9-26.

Feng, Qi, J George Shanthikumar. 2023. The framework of parametric and nonparametric operational data analytics. *Production and Operations Management*, 32 (9), 2685-2703.

Harsha, Pavithra, Ramesh Natarajan, Dharmashankar Subramanian. 2021. A prescriptive machine-learning framework to the price-setting newsvendor problem. *INFORMS Journal on Optimization*, 3 (3), 227-253.

Kallus, Nathan, Xiaojie Mao. 2022. Stochastic optimization forests. *Management Science*, 69 (4), 1975-1994. doi:10.1287/mnsc.2022.4458. URL https://doi.org/10.1287/mnsc.2022.4458.

Kannan, Rohit, Guzin Bayraksan, James R. Luedtke. 2022. Data-driven sample average approximation with covariate information.

Lin, Shaochong, Youhua Chen, Yanzhi Li, Zuo-Jun Max Shen. 2022. Data-driven newsvendor problems regularized by a profit risk constraint. *Production and Operations Management*, 31 (4), 1630-1644.

Liu, Junyi, Guangyu Li, Suvrajeet Sen. 2021. Coupled learning enabled stochastic programming with endogenous uncertainty. *Mathematics of Operations Research*, 47 (2), 1681-1705.

Luo, Fengqiao, Sanjay Mehrotra. 2020. Distributionally robust optimization with decision dependent ambiguity sets. *Optimization Letters*, 14 (8), 2565-2594. doi:10.1007/s11590-020-01574-3. URL https://doi.org/10.1007/s11590-020-01574-3https://link.springer.com/content/pdf/10.1007/s11590-020-01574-3.pdf.

Mendler-Dünner, Celestine, Juan Perdomo, Tijana Zrnic, Moritz Hardt. 2020a. Stochastic optimization for performative prediction. *International Conference on Machine Learning*, 7599-7609.

Mendler-Dünner, Celestine, Juan C. Perdomo, Tijana Zrnic, Moritz Hardt. 2020b. Performative prediction. *arXiv:2006.06887*, .

Noyan, Nilay, Gábor Rudolf, Miguel Lejeune. 2021. Distributionally robust optimization under a decision-dependent ambiguity set with applications to machine scheduling and humanitarian logistics. *INFORMS Journal on Computing*, 34 (2), 729-751. doi:10.1287/ijoc.2021.1096. URL https://doi.org/10.1287/ijoc.2021.1096.

Oroojlooyjadid, Afshin, Lawrence V Snyder, Martin Takáč. 2020. Applying deep learning to the newsvendor problem. *IISE Transactions*, 52 (4), 444-463.

Perdomo, Juan, Tijana Zrnic, Celestine Mendler-Dünner, Moritz Hardt. 2020. Performative prediction. *International Conference on Machine Learning*. PMLR, 7599-7609.

Rubner, Yossi, Carlo Tomasi, Leonidas J. Guibas. 2000. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40 (2), 99-121.

Sadana, Utsav, Abhilash Chenreddy, Erick Delage, Alexandre Forel, Emma Frejinger, Thibaut Vidal. 2023. A survey of contextual optimization methods for decision making under uncertainty.

Srivastava, P.R., Yijie Wang, Grani Adiwena Hanasusanto, Chin Pang Ho. 2021. On data-driven prescriptive analytics with side information: A regularized nadaraya-watson approach.

Zhang, Yanfei, Junbin Gao. 2017. Assessing the performance of deep learning algorithms for newsvendor problem. *Neural Information Processing: 24th International Conference, ICONIP 2017, Guangzhou, China, November 14-18, 2017, Proceedings, Part I 24*. Springer, 912-921.

E-Companion for Tackling Decision Dependency in Contextual Stochastic Optimization

EC.1 Description of Solution Methods

In this section, we explain the solution methods adopted in the numerical experiment section.

EC.1.1 Diminishing Step

The diminishing step adopt the step size η^r such that $\eta^r > \eta^{r+1}$ and $\sum_{r=0}^{\infty} \eta^r = \infty$. A typical choice is $\eta^r = C/(r+1)$, where C is a constant that can be adjusted to suit different problems.

EC.1.2 Armijo Step

Let $f(\cdot)$ denote the objective function we want to minimize. The Armijo principle chooses the step size η^r by the following steps (we denote the ascent direction as d^r) in algorithm 2

Algorithm 2 Armijo step size

Input: iteration solution x^r, contextual information z, α_0, $\beta \in (0,1)$, $\sigma \in [0,1)$, tolerance ε.

Output: step size η^r.

1: $\eta^r = \alpha_0$
2: $x^{r+1} = x^r + \eta^r d^r$;
3: **while** $\eta^r \geq \varepsilon$ and $f(x^r) - f(x^{r+1}) < \sigma\eta^r(\hat{G}_N(x^r;z))^T d^r$ **do**
4: $\quad \eta^r = \eta^r * \beta$;
5: $\quad x^{r+1} = x^r + \eta^r d^r$;
6: **end while**
7: **return** η^r

Note that the hyperparameter σ can be 0 in our problem. When $\sigma = 0$, the armijo step size ensure that the objective function descent in an approximate context. We also show the special meaning when $\sigma = 0$ in Proposition 4.

EC.2 Proofs

Proof of Lemma 1

The proof Proposition 1 roughly follows the proof of Theorem EC.9 in Bertsimas and Kallus (2019). However, there are difference between them since Proposition 1 is about the convergence of derivative function rather than the objective function.

Specifically, for every x, the marginal distribution of $y \sim f(y;x,z)$ is independent of y conditioned on z, the ignorability assumption satisfies. Furthermore, The feasible region for x is nonempty, and we only restrict the up and down limit of the two decisions.

Therefore, we need to prove that the expected gradient $\mathbb{E}[\nabla_x l(x,y)|x=x',z=z']$ is bounded and equicontinuous on x. First, from Assumption 4(c) we have $|\nabla_x l(x,y)| < \infty$ for every $x \in X$ and $y \in Y$, thus $\liminf_{x\in X,\|x\|\to\infty} \inf_{y\in Y} |\nabla_x l(x,y)| < \infty$. Then from Assumption 5, for any $x \in X, \varepsilon > 0, x's.t.\|x-x'\| \le \varepsilon/L_{1g}$,

$$\begin{aligned}\|\nabla_x l(x,y) - \nabla_x l(x',y')\| &\le L_{1g}\|x'-x\| \\ &\le \varepsilon.\end{aligned}$$

Thus $\nabla_x l(x,y)$ is equicontinuous. Then the proof is completed by Theorem EC.9 in Bertsimas and Kallus (2019).

Proof of Proposition ??

Assume that Assumption 1 holds. We rewrite the objective expectation to the integrate form:

$$\nabla_x \mathbb{E}_{f(y;x,z)}[l(x,y)] = \nabla_x \int_{y\in\Omega} l(x,y) f(y;x,z) dy.$$

Suppose that the derivative of $l(x,y), f(y;x,z)$ can be bounded by an L^1 function g for all x,y, then the derivative and integration operator can be switched.

$$\begin{aligned}\nabla_x \mathbb{E}_{f(y;x,z)}[l(x,y)] &= \int_{y\in\Omega} \nabla_x [l(x,y) f(y;x,z)] dy \\ &= \int_{y\in\Omega} (\nabla_x l(x,y))\, f(y;x,z) dy \\ &\quad + \int_{D\in\Omega} (\nabla_x f(y;x,z))\, l(x,y) dy \\ &= \mathbb{E}_{D\sim f_D(p,z)}[\partial_{p,q} l(x,y)] \\ &\quad + \int_{D\in\Omega} (\nabla_x f(y;x,z))\, l(x,y) dy.\end{aligned}$$

Therefore, the equality holds only when the second term of the last equation equals to 0, which is not guaranteed. So the expectation of cost gradient do not equal to the gradient of objective expectation and thus the convergence of approximate gradient fails.

Before we begin to proof the convergence results, we first state some important results. The following lemmas show how Assumption 3 affects the distance between expectations of different distributions.

LEMMA EC.1. *Kantorovich-Rubinstein For all function f that is $1-$Lipschitz*

$$\|\mathbb{E}_{d\sim D(p)}\mathbb{E}[f(d)] - \mathbb{E}_{d\sim D(p')}\mathbb{E}[f(d)]\| \le W_1(D(p), D(p')).$$

LEMMA EC.2. *Suppose Assumption 3 holds. Let $f : R^n \to R^d$ be an $L-$Lipschitz function, and let $X, X' \in \mathbb{R}^n$ be random variables such that $W_1(X,X') \le C$. Then*

$$\|\mathbb{E}[f(X)] - \mathbb{E}[f(X')]\|_2 \le LC. \tag{EC.1}$$

Proof of Lemma EC.2

Since

$$\begin{aligned}\|\mathbb{E}[f(X)]-\mathbb{E}[f(X')]\|_2^2 &= (\mathbb{E}[f(X)]-\mathbb{E}[f(X')])^T(\mathbb{E}[f(X)]-\mathbb{E}[f(X')]) \\ &= \|\mathbb{E}[f(X)]-\mathbb{E}[f(X')]\|_2 \frac{(\mathbb{E}[f(X)]-\mathbb{E}[f(X')])^T}{\|\mathbb{E}[f(X)]-\mathbb{E}[f(X')]\|_2}(\mathbb{E}[f(X)]-\mathbb{E}[f(X')]),\end{aligned}$$

we define the unit vector $e := \frac{(\mathbb{E}[f(X)]-\mathbb{E}[f(X')])^T}{\|\mathbb{E}[f(X)]-\mathbb{E}[f(X')]\|_2}$, we can get:

$$\|\mathbb{E}[f(X)]-\mathbb{E}[f(X')]\|_2^2 = \|\mathbb{E}[f(X)]-\mathbb{E}[f(X')]\|_2(\mathbb{E}[e^T f(X)]-\mathbb{E}[f(X')]).$$

Since f is a one-dimensional $L-$lipschitz function, we can apply Lemma EC.1 and Assumption 3 to obtain that for all e,

$$\|\mathbb{E}[f(X)]-\mathbb{E}[f(X')]\|_2^2 \le \|\mathbb{E}[f(X)]-\mathbb{E}[f(X')]\|_2 LC.$$

Thus completing the proof

Proof of Theorem 2

We analyze the error of x_N^{k+1}:

$$\begin{aligned}\|x_N^{k+1}-x^*\|^2 =& \|x_N^k-\eta^k\hat{G}_N(x_N^k,z)-x^*\|^2 \\ =& \|x_N^k-x^*\|^2 - 2\eta^k\hat{G}_N(x_N^k,z)^T(x_N^k-x^*)+(\eta^k)^2\|\hat{G}_N(x_N^k,z)\|^2.\end{aligned}$$

let $N\to+\infty$ for both sides, denote $\lim_{N\to\infty}x_N^k$ as x^k for simplicity. From Proposition 1 we have $\lim_{N\to\infty}\hat{G}_N(x,z)=\mathbb{E}_{f(y;x,z)}[\nabla_x l(x,y)]$ for any x. Therefore, for any $\zeta>0, x, z$ and vector v, $\exists N_0$, $\forall N>N_0$,

$$\|\hat{G}_N(x,z)\| \le \|\mathbb{E}_{f(y;x,z)}[\nabla_x l(x,y)]\|+\zeta,$$

and

$$\hat{G}_N(x,z)^T v \le \mathbb{E}_{f(y;x,z)}[\nabla_x l(x,y)]^T v + \zeta\|v\|.$$

Thus, for any $\zeta>0$, we let $\zeta_1=\frac{\zeta}{\|x_N^k-x^*\|}$ and $\zeta_2=\sqrt{\zeta/\eta^k}$, $\exists N_0$, for any fix $N>N_0$ we have:

$$\begin{aligned}\|x_N^{k+1}-x^*\|^2 =& \|x_N^k-x^*\|^2 - 2\eta^k\mathbb{E}_{f(y;x_N^k,z)}[\nabla_x l(x_N^k,y)]^T(x_N^k-x^*) \\ &+(\eta^k)^2\|\mathbb{E}_{f(y;x_N^k,z)}[\nabla_x l(x_N^k,y)]\|^2+3\eta^k\zeta.\end{aligned}$$

We bound the second term by convexity of the cost function

$$\begin{aligned}\mathbb{E}_{f(y;x_N^k,z)}[\nabla_x l(x_N^k,y)]^T(x_N^k-x^*) =& \mathbb{E}_{f(y;x_N^k,z)}[\nabla l(x_N^k,y)^T(x_N^k-x^*)] \\ \ge& \mathbb{E}_{f(y;x_N^k,z)}[l(x_N^k,y)-l(x^*,y)].\end{aligned}$$

For the third term, we bound by Assumption 4.

$$\|\mathbb{E}_{f(y;x_N^k,z)}[\nabla_x l(x_N^k,y)]\|^2 \le L_3^c.$$

Thus

$$2\eta^k \mathbb{E}_{f(y;x_N^k,z)}[l(x_N^k,y)-l(x^*,y)] \le -\|x_N^{k+1}-x^*\|^2+\|x_N^k-x^*\|^2+(\eta^k)^2(L_3^c)^2+3\eta^k\zeta.$$

We further investigate the right side. We have

$$\begin{aligned}
\mathbb{E}_{f(y;x_N^k,z)}[l(x_N^k,y)-l(x^*,y)] =& -\mathbb{E}_{f(y;x_N^k,z)}[l(x^*,y)]+\mathbb{E}_{f(y;x^*,z)}[l(x^*,y)] \\
& -\mathbb{E}_{f(y;x^*,z)}[l(x^*,y)]+\mathbb{E}_{f(y;x_N^k,z)}[l(x_N^k,y)] \\
\ge& -|\mathbb{E}_{f(y;x_N^k,z)}[l(x^*,y)]-\mathbb{E}_{f(y;x^*,z)}[l(x^*,y)]| \\
& -\mathbb{E}_{f(y;x^*,z)}[l(x^*,y)]+\mathbb{E}_{f(y;x_N^k,z)}[l(x_N^k,y)] \\
\ge& -\varepsilon L_2\|x^*-x_N^k\|-\mathbb{E}_{f(y;x^*,z)}[l(x^*,y)]+\mathbb{E}_{f(y;x_N^k,z)}[l(x_N^k,y)].
\end{aligned}$$

This inequality reflects the main difficulty of our proof: to construct the gap of $g(x)=\mathbb{E}_{f(y;x,z)}[l(x,y)]$ between x^* and x_N^k. Then we substitute the inequality and have

$$\begin{aligned}
2\eta^k(\mathbb{E}_{f(y;x_N^k,z)}[l(x_N^k,y)]-\mathbb{E}_{f(y;x^*,z)}[l(x^*,y)]) \le& 2\eta^k\varepsilon L_2\|x^*-x_N^k\|-\|x_N^{k+1}-x^*\|^2 \\
& +\|x_N^k-x^*\|^2+(\eta^k L_3^c)^2+3\eta^k\zeta.
\end{aligned}$$

Take summation from $r=0$ to k and take the minimum of the left side, we obtain

$$\begin{aligned}
(2\sum_{r=0}^k\eta^r)\min_{0\le r\le k}\{\mathbb{E}_{f(y;x^r,z)}[l(x^r,y)]-\mathbb{E}_{f(y;x^*,z)}[l(x^*,y)]\} \le& 2\varepsilon L_2\sum_{r=0}^k\eta^r\|x^*-x^k\| \\
& +\|x^0-x^k\|^2+(L_3^c)^2\sum_{r=0}^k(\eta^r)^2+3\sum_{r=0}^k\eta^r\zeta.
\end{aligned}$$

Hence we complete the proof by dividing $2\sum_{r=0}^k\eta^r$ on both sides.

~~Proof of Proposition 2~~

~~We investigate the distance between x_N^k and a stable point x_{PS}.~~

$$\begin{aligned}
\|x_N^{k+1}-x_{PS}\|^2 =& \|x_N^k-\eta\hat{G}_N(x_N^k;z)-x_{PS}\|^2 \\
=& \|x_N^k-x_{PS}\|^2-2\eta\hat{G}_N(x_N^k)^T(x_N^k-x_{PS})+(\eta^2)\|\hat{G}_N(x_N^k)\|^2.
\end{aligned}$$

~~We begin by upper bounding the second term. From Proposition 1, we know that for any $\xi>0$, there exists a sample size N_0 such that $\sup_x\|\hat{G}_N(x)-\mathbb{E}_{f(y;x,z)}[\nabla_x l(x,y)]\|\le\xi$ for all $N>N_0$. Thus we have~~

$$\hat{G}_N(x_N^k)^T(x_N^k-x_{PS}) \ge \mathbb{E}_{f(y;x_N^k,z)}[\nabla_x l(x_N^k,y)]^T(x_N^k-x_{PS})-\xi\|x_N^k-x_{PS}\|.$$

~~We can further bound the second term using the same approach as the proof of proposition 2.3 in Mendler-Dünner et al. (2020b)'s work. They give that~~

$$\mathbb{E}_{f(y;x_N^k,z)}[\nabla_x l(x_N^k,y)]^T(x_N^k-x_{PS}) \ge A\|x_N^k-x_{PS}\|^2.$$

We then bound the third term:

$$\|\hat{G}_N(x_N^k)\|^2 \le \xi^2 + \|\mathbb{E}_{f(y,x_N^k,z)}[\nabla_x l(x_N^k, y)]\|^2.$$

We can also adopt the same approach in the proof of proposition 2.3 in Mendler-Dünner et al. (2020b). They give that under Assumptions 4 and 3,

$$\|\mathbb{E}_{f(y,x_N^k,z)}[\nabla_x l(x_N^k, y)]\|^2 \le 2B^2\|x_N^k - x_{PS}\|^2.$$

Therefore, we obtain

$$\|x_N^{k+1} - x_{PS}\|^2 \le (1 - 2\eta A + 2\eta^2 B^2)\|x_N^k - x_{PS}\|^2 + 2\eta\xi\|x_N^k - x_{PS}\| + \xi^2\eta^2. \tag{EC.2}$$

In case 1, to give a reasonable distance bound, we need to choose η such that the right-hand side is a perfect quadratic polynomial. Thus we choose η such that

$$4B^2\eta^2 - 2A\eta + 1 = 0.$$

Note that from the Viete's theorem, the two solutions are both positive since we assume $A > 0$. Thus we only need to ensure that the equation have real solution, that is

$$4A^2 \ge 16B^2, \quad A \ge 2B.$$

And we take the square root two both sides of equation (EC.4)

$$\|x_N^{k+1} - x_{PS}\| \le \sqrt{1 - 2\eta A + 2\eta^2 B^2}\|x_N^k - x_{PS}\| + \xi\eta.$$

We denote $C = \sqrt{1 - 2\eta A + 2\eta^2 B}$ and divide both sides by C^{k+1}

$$\frac{\|x_N^{k+1} - x_{PS}\|}{C^{k+1}} \le \frac{\|x_N^k - x_{PS}\|}{C^k} + \frac{\xi\eta}{C^{k+1}}.$$

Take the summation on both sides from 0 to $k+1$ and we obtain

$$\|x_N^{k+1} - x_{PS}\| \le C^{k+1}\|x_N^1 - x^*\| + \xi\eta\frac{1 - C^{k+1}}{1 - C}.$$

Note that $\eta A - \eta^2 B^2 = \frac{2\eta A + 1}{4} > 0$, thus $C < 1$ and the distance is decreasing.

Now we focus on case 2. Since the quadratic term on the right-hand side of (EC.4) is less than zero, we obtain

$$\|x_N^{k+1} - x_{PS}\|^2 \le 2\eta\xi\|x_N^k - x_{PS}\| + \xi^2\eta^2. \tag{EC.3}$$

Thus

$$\|x_N^{k+1} - x_{PS}\|^2 - \|x_N^k - x_{PS}\|^2 \le \|x_N^k - x_{PS}\|^2 + 2\eta\xi\|x_N^k - x_{PS}\| + \xi^2\eta^2.$$

If $\|x_N^k - x_{PS}\| \ge (1 + \sqrt{2})\xi\eta$, we can derive that $\|x_N^{k+1} - x_{PS}\|^2 - \|x_N^k - x_{PS}\|^2 \le 0$, which indicates that although the distance may exceed the bound $(1 + \sqrt{2})\xi\eta$ some time, it will decrease immediately until it reach the bound, hence complete the proof.

EC.2.1 Proof of Proposition 1

We investigate the distance between x_N^k and a stable point x_{PS}.

$$\begin{aligned}\|x_N^{k+1} - x_{PS}\|^2 =& \|x_N^k - \eta \hat{G}_N(x_N^k; z) - x_{PS}\|^2 \\ =& \|x_N^k - x_{PS}\|^2 - 2\eta \hat{G}_N(x_N^k)^T (x_N^k - x_{PS}) + (\eta^2)\|\hat{G}_N(x_N^k)\|^2.\end{aligned}$$

We begin by upper bounding the second term. From Proposition 1, we know that for any $\xi > 0$, there exists a sample size N_0 such that $\sup_x \|\hat{G}_N(x) - \mathbb{E}_{f(y;x,z)}[\nabla_x l(x,y)]\| \le \xi$ for all $N > N_0$. Thus we have

$$\hat{G}_N(x_N^k)^T (x_N^k - x_{PS}) \ge \mathbb{E}_{f(y;x_N^k,z)}[\nabla_x l(x_N^k, y)]^T (x_N^k - x_{PS}) - \xi \|x_N^k - x_{PS}\|.$$

We can further bound the second term using the same approach as the proof of Proposition 2.3 in Mendler-Dünner et al. (2020b). They give that

$$\mathbb{E}_{f(y;x_N^k,z)}[\nabla_x l(x_N^k, y)]^T (x_N^k - x_{PS}) \ge A \|x_N^k - x_{PS}\|^2.$$

We then bound the third term:

$$\|\hat{G}_N(x_N^k)\|^2 \le (\xi + \|\mathbb{E}_{f(y;x_N^k,z)}[\nabla_x l(x_N^k, y)]\|)^2.$$

We can also adopt the same approach in the proof of Proposition 2.3 in Mendler-Dünner et al. (2020b). They give that under Assumptions 4 and 3,

$$\|\mathbb{E}_{f(y;x_N^k,z)}[\nabla_x l(x_N^k, y)]\|^2 \le 2B^2 \|x_N^k - x_{PS}\|^2.$$

Therefore, we obtain

$$\|x_N^{k+1} - x_{PS}\|^2 \le (1 - 2\eta A + 2\eta^2 B^2)\|x_N^k - x_{PS}\|^2 + (2\eta\xi + 2\sqrt{2}\xi\eta^2 B)\|x_N^k - x_{PS}\| + \xi^2\eta^2. \tag{EC.4}$$

Denote $\|x_N^k - x_{PS}\| = X_k \ge 0$, we have

$$X_{k+1}^2 \le (1 - 2\eta A + 2\eta^2 B^2) X_k + (2\eta\xi + 2\sqrt{2}\xi\eta^2 B) X_k + \xi^2\eta^2 \tag{EC.5}$$

Denote $h(x) = (1 - 2\eta A + 2\eta^2 B^2)x + (2\eta\xi + 2\sqrt{2}\xi\eta^2 B)x + \xi^2\eta^2$, $X_{k+1}^2 \le h(X_K)$. *Now that* $1 - 2\eta A + 2\eta^2 B^2 > 0$, *and* $\eta\xi > 0$, *as a quadratic function we can easily get* $h(x)$ *is monotonically increasing in* $[0, +\infty)$.

Denote $g(x) = h(x) - x^2 = (-2\eta A + 2\eta^2 B^2)x + (2\eta\xi + 2\sqrt{2}\xi\eta^2 B)x + \xi^2\eta^2$. *Now that* $1 - 2\eta A + 2\eta^2 B^2 < 1$, *we have* $-2\eta A + 2\eta^2 B^2 < 0$. $g(x)$ *as a quadratic function has two solutions:*

$$\begin{aligned} X_1 &= \frac{\xi}{2(A - \eta B^2)}(1 + \sqrt{2}\eta B - \sqrt{-2B^2\eta^2 + 2A\eta + (1 + \sqrt{2}\eta B)^2}) < 0, \\ X_2 &= \frac{\xi}{2(A - \eta B^2)}(1 + \sqrt{2}\eta B + \sqrt{-2B^2\eta^2 + 2A\eta + (1 + \sqrt{2}\eta B)^2}) > 0 \end{aligned} \tag{EC.6}$$

When $x \in [0, X_2), g(x) > 0$; *When* $x \in (X_2, +\infty), g(x) < 0$.

Then the series $\{X_k\}$ *can be separated into two cases:*

Case 1. There exists a certain K_1, $X_{K_1} \le X_2$:

$$X_{K_1+1}^2 \le h(X_{K_1}) \le h(X_2) = g(X_2) + X_2^2 = X_2^2$$

Namely, $X_{K_1+1} \le X_2$. *By deduction,* $\forall k > K_1, X_k \le X_2 < X_2 + \varepsilon$.

Case 2. For all $k > 0$, $X_k > X_2$: *In this case,* $X_{k+1}^2 - X_K^2 \le g(X_K) < 0$. *Namely* $X_{k+1} < X_k$, *series* $\{X_k\}$ *is monotonically decreasing. Since series* $\{X_k\}$ *has lower bound* X_2, *it has infimum. Denote* $\inf_{k \in N_+} X_k = U \ge X_2$.

Case 2.1 $U = X_2$: *From definition of infimum,* $\forall \varepsilon > 0$, $\exists K_2 > 0$, $X_{K_2} \le U + \varepsilon$. *And thus* $\forall k > K_2, X_k < X_{K_2} \le X_2 + \varepsilon$.

Case 2.2 $U > X_2$: *From definition of infimum,* $\forall \varepsilon > 0$, $\exists K_3 > 0$, $X_{K_3}^2 \le U^2 + \varepsilon$.

$$X_{K_3+1}^2 \le h(X_{K_3}) < h(\sqrt{U^2 + \varepsilon}) = g(\sqrt{U^2 + \varepsilon}) + U^2 + \varepsilon$$

Let $K(\varepsilon) = g(\sqrt{U^2 + \varepsilon}) + U^2 + \varepsilon$, $K(\varepsilon)$ *is obviously continuous.*

$$\lim_{\varepsilon \to 0} K(\varepsilon) = h(0) = g(U) + U^2 < U^2 (\forall x > X_2, g(x) < 0)$$

And thus there exists $\delta > 0$, $\forall 0 < \varepsilon < \delta, h(\varepsilon) < U^2$. *Namely there exists* $\varepsilon > 0$, $X_{K_3+1}^2 \le h(X_{k_3}) < K(\varepsilon) < U^2$. *But* $X_{K_3+1} < U$ *contradicts with the case assumption that* $\forall X_k, X_k \ge U > X_2$. *Hence this case doesn't exist.*

As a conclusion, $\exists K_0 = \max\{K_1, K_2\} > 0$, $\forall k > K_0, X_k < X_2 + \varepsilon$ *and the proof is completed.*

Proof of Theorem 3

Since $l(x, y)$ is strongly convex in x and L_2-Lipschitz continuous in y, the proof is then complete by imposing the triangular inequality to Lemma 2 and Proposition 1.

Proof of Proposition 3

The proof is divided into two steps. In the first step, we prove that the objective function $\mathbb{E}_{f(y;x,z)}[l(x,y)]$ has Lipschitz gradient in x. Then we prove that under diminishing step, any converging subsequence converge to the stationary point.

We denote $g(x) = \mathbb{E}_{f(y;x,z)}[l(x,y)]$, then for any $x_1, x_2 \in \mathcal{X}$

$$\|\nabla_x g(x_1) - \nabla_x g(x_2)\| = \|\nabla_x \mathbb{E}_{f(y;x_1,z)}[l(x_1,y)] - \nabla_x \mathbb{E}_{f(y;x_2,z)}[l(x_2,y)]\|.$$

According to Assumption 2, we can write the expectation to integrate form and change the integrate operator and derivative operator.

$$\begin{aligned}
\|\nabla_x g(x_1) - \nabla_x g(x_2)\| =& \|\int_{y\in\Omega} \nabla_x (l(x_1,y) f(y;x_1,z))dy - \int_{y\in\Omega} \nabla_x (l(x_2,y) f(y;x_2,z))dy\| \\
\leq& \|\int_y l(x_1,y)(\nabla_x f(y;x_1,z)) - l(x_2,y)(\nabla_x f(y;x_2,z))dy\| \\
&+ \|\int_y (\nabla_x l(x_1,y)) f(y;x_1,z) - (\nabla_x l(x_2,y)) f(y;x_2,z)dy\| \\
=& I + II.
\end{aligned}$$

The second inequality follows by the multiplication rule of derivative. We then analyze I and II respectively.

$$\begin{aligned}
I \leq& \|\int_y l(x_1,y)\nabla_x f(y;x_1,z)dy - \int_y l(x_1,y)\nabla_x f(y;x_2,z)dy\| \\
&+ \|\int_y l(x_1,y)\nabla_x f(y;x_2,z)dy - \int_y l(x_2,y)\nabla_x f(y;x_2,z)dy\| \\
\leq& \int_y |l(x_1,y)| \|\nabla_x f(y;x_1,z) - \nabla_x f(y;x_2,z)\| dy \\
&+ \int_y |l(x_1,y) - l(x_2,y)| \|\nabla_x f(y;x_2,z)\| dy \\
\leq& S_\Omega L_4 L_{3g} \|x_1 - x_2\| + S_\Omega L_5 L_1 \|x_1 - x_2\|.
\end{aligned}$$

The first inequality holds from the triangular inequality. The second inequality holds by the Cauchy-Schwarz inequality. The third inequality holds by the Lipschitz continuous characteristic and intermediate value theorem, where S_Ω denotes of the measurement of the set Ω.

We can also bound the second term by the following steps:

$$\begin{aligned}
II \leq& \|\int_y \nabla_x l(x_1,y) f(y;x_1,z)dy - \int_y \nabla_x l(x_1,y) f(y;x_2,z)dy\| \\
&+ \|\int_y \nabla_x l(x_1,y) f(y;x_2,z)dy - \int_y \nabla_x l(x_2,y) f(y;x_2,z)dy\| \\
=& \|\mathbb{E}_{f(y;x_1,z)}[\nabla_x l(x_1,y)] - \mathbb{E}_{f(y;x_2,z)} \nabla_x l(x_1,y)\| + \|\mathbb{E}_{f(y;x_2,z)}[\nabla_x l(x_1,y) - \nabla_x l(x_1,y)]\| \\
\leq& \varepsilon L_{2g} \|x_1 - x_2\| + L_{2g} \|x_1 - x_2\|.
\end{aligned}$$

The first inequality holds by the triangular inequality. The first equality holds by the definition of expectation. The second inequality holds by Lemma EC.2 and the definition of Lipschitz gradient.

Thus, $\|\nabla_x g(x_1) - \nabla_x g(x_2)\| \leq [(\varepsilon+1)L_{2g} + S_\Omega(L_1L_5 + L_4L_{3g})]\|x_1 - x_2\|$. Hence the objective function has Lipschitz gradient and $L = (\varepsilon+1)L_{2g} + S_\Omega(L_1L_5 + L_4L_{3g})$.

recall that the update rule is given by

$$x_N^{r+1} = x_N^r + \eta^r \hat{G}_N(x;z).$$

From descent lemma, we have

$$g(x_N^{r+1}) \leq g(x_N^r) + \eta^r \hat{G}_N(x_N^r;z)^T \nabla g(x_N^r) + \frac{L(\eta^r)^2}{2}\|\hat{G}_N(x_N^r;z)\|^2.$$

Taking $N \to \infty$ on both sides, since $g(x)$ is continuous and $\lim_{N\to\infty} \hat{G}_N(x;z) = \mathbb{E}_{f(y;x,z)}[\nabla_x l(x,y)]$ from Proposition 1.

$$g(x^{r+1}) - g(x^r) \leq \eta^r \mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)]^T \nabla g(x^r) + \frac{L(\eta^r)^2}{2}\|\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)]\|^2.$$

where $x^r = \lim_{N\to\infty} x_N^r$.

Since

$$\begin{aligned}\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)]^T g(x^r) =& \|\nabla_x \mathbb{E}_{f(y;x^r,z)}[l(x^r,y)]\|^2 + \|\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)]\|^2 \\ &- \|\int_y l(x^r,y)\nabla_x f(y;x^r,z)dy\|^2 \\ \geq& (\|\nabla_x \mathbb{E}_{f(y;x^r,z)}[l(x^r,y)]\|^2 - L_4^2 L_5^2 S_\Omega^2) + \|\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)]\|^2.\end{aligned}$$

Note that since the range of y is limited, we can scale the random parameters y so that $S_\Omega^2 \leq \frac{\|\nabla_x \mathbb{E}_{f(y;x^r,z)}[l(x^r,y)]\|}{L_4L_5}$. Thus,

$$\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)]^T g(x^r) \geq \|\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)]\|^2.$$

Therefore,

$$g(x^{r+1}) - g(x^r) \leq -\eta^r(1 - \frac{L\eta^r}{2})\|\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)]\|^2.$$

Since η^r is diminishing, for any $\xi \in (0,1)$, there exists $\bar{r}$ such that for any $r \geq \bar{r}$, we have

$$g(x^{r+1}) - g(x^r) \leq -\eta^r \xi \|\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)]\|^2.$$

Since $\lim_{r\in\mathcal{K}\to\infty} x^r = \bar{x}$ and $g(x)$ is continuous, we have $\lim_{r\to\infty} g(x^r) = g(\bar{x})$. Taking summation on both sides from $r = \bar{r}$ to ∞, we can obtain that

$$\sum_{r=\bar{r}}^{\infty} \eta^r \xi \|\mathbb{E}_{f(D;x^r,z)}[\nabla_x l(x^r,D)]\|^2 \leq g(x^{\bar{r}}) - \lim_{r\to\infty} g(x^r).$$

Since $\sum_{r=\bar{r}}^{\infty} \eta^r = +\infty$, we have $\lim_{r\in\mathcal{K}\to\infty} \|\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)]\|^2 = 0$, hence $\mathbb{E}_{f(y;\bar{x},z)}[\nabla_x l(\bar{x},y)] = 0$ and the proof is completed.

Proof of Proposition 4

To simplify the denotation, we omit the limitation of $N \to \infty$. Therefore the descent direction is $d^r = -\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)]$, where $x^r = \lim_{N\to\infty} x_N^r$. According to the armijo principle:

$$g(x^r) - g(x^{r+1}) \geq -\eta^r \sigma \|\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)]\|^T d^r.$$

Since $\lim_{r(\in\mathcal{K})\to\infty} \sup_r \|\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)]\| \geq 0$. The sequence $\mathbb{E}_{f(y;x^r,z)}[l(x^r,y)]$ decreases monotonically and have a lower bound. Thus

$$\lim_{r(\in\mathcal{K})\to\infty} g(x^r) - g(x^{r+1}) = 0,$$

which is followed by

$$\lim_{r(\in\mathcal{K})\to\infty} \eta^r = 0.$$

Hence, by the definition of the armijo rule, we must have for some index $\bar{r} \geq 0$

$$g(x^r) - g(x^r + \frac{\eta^r}{\beta} d^r) < -\sigma \frac{\eta^r}{\beta} \|\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)]\|^T d^r, \forall r \in \mathcal{K}, r \geq \bar{r}.$$

We denote

$$p^r := \frac{d^r}{\|d^r\|}, \quad \bar{\eta}^r := \frac{\eta^r \|d^r\|}{\beta}.$$

Since $\|p^r\| = 1$, there exists a subsequence $\{p^r\}_{\bar{\mathcal{K}}}$ of $\{p^r\}_{\mathcal{K}}$ such that $\{p^r\}_{\bar{\mathcal{K}}} \to \bar{p}$, where $\bar{p}$ is a unit vector. Then

$$\frac{g(x^r) - g(x^{r+1})}{\bar{\eta}^r} < -\sigma(\mathbb{E}_{f(x^r;y,z)}[\nabla_x l(y,x^r)])^T p^r.$$

Hence,

$$\frac{g(x^r) - \mathbb{E}_{f(y;x^r,z)}[l(x^{r+1},y)] + \mathbb{E}_{f(y;x^r,z)}[l(x^{r+1},y)] - g(x^{r+1})}{\bar{\eta}^r} < -\sigma(\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)])^T p^r. \tag{EC.7}$$

By Lemma EC.2, $g(x^r) - g(x^{r+1}) \geq -\varepsilon L_1 \|\bar{\eta}^r p^r\|$.

By using the mean value theorem,

$$\begin{aligned} \frac{-\varepsilon L_1 \|\bar{\eta}^r p^r\|}{\bar{\eta}^r} &+ \mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r + \tilde{\alpha}^r p^r, y)]^T p^r \\ &< -\sigma(\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)])^T p^r. \end{aligned} \tag{EC.8}$$

Let $r(\in \bar{\mathcal{K}}) \to \infty$,

$$-\varepsilon L_1 - (\mathbb{E}_{f(y;\bar{x},z)}[\nabla_x l(\bar{x},y)])^T \bar{p} < -\sigma \mathbb{E}_{f(y;\bar{x},z)}[\nabla_x l(\bar{x},y)]^T \bar{p}.$$

Substituting $d^r = -\mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x^r,y)]$, we have

$$-\varepsilon L_1 < -(1-\sigma)\|\mathbb{E}_{f(y;\bar{x},z)}[\nabla_x l(\bar{x},y)]\|.$$

which completes the proof.

Proof of Proposition 5

For any given sequence $\{x_N^r\}_{r=1}^k$, denote $g_r(x) = \mathbb{E}_{f(y;x^r,z)}[l(x,y)]$. Then $\nabla_x g_r(x) = \mathbb{E}_{f(y;x^r,z)}[\nabla_x l(x,y)]$.

Since $l(x,y)$ has $L_{1g}-$Lipschitz gradient, by the descent lemma, when $N \to \infty$,

$$\begin{aligned} g_r(x_N^{r+1}) - g_r(x_N^r) =& g_r(x_N^r - \eta \mathbb{E}_{f(y;x_N^r,z)}[\nabla_x l(x_N^r, y)]) - g_r(x_N^r) \\ \leq& -(1 - \frac{L_{1g}\eta}{2})\eta \|\mathbb{E}_{f(y;x_N^r,z)}[\nabla_x l(x_N^r, y)]\|^2. \end{aligned}$$

Thus,

$$\sum_{r=0}^{k} \|\mathbb{E}_{f(y;x_N^r,z)}[\nabla_x l(x_N^r, y)]\|^2 \leq \frac{2}{\eta} \sum_{r=0}^{k} \left(g_r(x_N^{r+1}) - g_r(x_N^r) \right). \tag{EC.9}$$

And,

$$\begin{aligned} \sum_{r=0}^{k} g_r(x_N^{r+1}) - g_r(x_N^r) =& \sum_{r=0}^{k} [g_{r+1}(x_N^{r+1}) - g_r(x_N^r)] + [g_r(x_N^{r+1}) - g_{r+1}(x_N^{r+1})] \\ \leq& g_0(x_N^0) - g_*(x^*) + \sum_{r=0}^{k} g_r(x_N^{r+1}) - g_{r+1}(x_N^{r+1}) \\ \leq& g_0(x_N^0) - g_*(x^*) + L_1 \varepsilon \sum_{r=1}^{k} \|x_N^r - x_N^{r-1}\| \\ =& g_0(x_N^0) - g_*(x^*) + L_1 \varepsilon \sum_{r=1}^{k} \|\eta \mathbb{E}_{f(y;x_N^r,z)}[\nabla_x l(x_N^r, y)]\| \\ \leq& g_0(x_N^0) - g_*(x^*) + L_1 \varepsilon \sum_{r=1}^{k} \eta L_3^c, \end{aligned}$$

where the first inequality holds because $g_*(x^*) \leq g_r(x_N^r)$ for any x^r, the second inequality holds by Lemma EC.2 and Assumption 3, and the third inequality holds by Assumption 4(c).

Then the proof completes by taking the union inequality on the left side of (EC.9) and dividing both sides by $k+1$.

Proof of Theorem 1

We proof the theorem by contradiction. Suppose x^* maximize $\max_x g(x) = \mathbb{E}_{f(y;x,z)}[l(x,y)]$, and $\|\mathbb{E}_{f(y;x^*,z)}[\nabla_x l(x^*, y)]\| > L_1 \varepsilon$. Then for any $x_1 \in X$,

$$\begin{aligned} g(x_1) - g(x^*) &= \left(\mathbb{E}_{f(y;x_1,z)}[l(x_1,y)] - \mathbb{E}_{f(y;x,z)}[l(x,y)] \right) \\ &= \left(\mathbb{E}_{f(y;x_1,z)}[l(x_1,y)] - \mathbb{E}_{f(y;x^*,z)}[l(x_1,y)] \right) \\ &\quad + \left(\mathbb{E}_{f(y;x^*,z)}[l(x_1,y)] - \mathbb{E}_{f(y;x,z)}[l(x,y)] \right). \end{aligned}$$

From Lemma EC.2, we have

$$|\mathbb{E}_{f(y;x_1,z)}[l(x_1,y)] - \mathbb{E}_{f(y;x^*,z)}[l(x_1,y)]| \leq L_1 \varepsilon \|x_1 - x^*\|.$$

For the second term, we expand $l(x_1,y)$ at x^* and obtain

$$\mathbb{E}_{f(y;x^*,z)}[l(x_1,y)] - \mathbb{E}_{f(y;x,z)}[l(x,y)] = \mathbb{E}_{f(y;x^*,z)}[\nabla_x l(x^*, y)]^T (x_1 - x^*) + o(\|x_1 - x^*\|),$$

where $o(\|x_1 - x^*\|)$ denotes the first-order infinitesimals to $\|x_1 - x^*\|$. By substituting the two terms above and divide both sides by $\|x_1 - x^*\|$, we obtain

$$\frac{g(x_1) - g(x^*)}{\|x_1 - x^*\|} \geq -L_1\varepsilon + \mathbb{E}_{f(y;x^*,z)}[\nabla_x l(x^*, y)]^T \frac{(x_1 - x^*)}{\|x_1 - x^*\|} + \frac{o(\|x_1 - x^*\|)}{\|x_1 - x^*\|}.$$

We let $x_1 - x^*$ take the same direction of $\mathbb{E}_{f(y;x^*,z)}[\nabla_x l(x^*, y)]$, hence the second term on the right side becomes $\|\mathbb{E}_{f(y;x^*,z)}[\nabla_x l(x^*, y)]\|$. Therefore, for any $\xi > 0$, there exists x_1 that is sufficiently close to x^* such that

$$\frac{g(x_1) - g(x^*)}{\|x_1 - x^*\|} \geq -L_1\varepsilon + \|\mathbb{E}_{f(y;x^*,z)}[\nabla_x l(x^*, y)]\| - \xi.$$

Since $\|\mathbb{E}_{f(y;x^*,z)}[\nabla_x l(x^*, y)]\| > L_1\varepsilon$ and ξ can be sufficiently small, we have $g(x_1) - g(x^*) > 0$, which contradicts with the condition that $g(x^*)$ is the optimal solution.

EC.3 Experiment Supplements

EC.3.1 Description of data

Real data The real dataset comes from a real-world power plant pricing scenario (https://www.kaggle.com/datasets/aramacus/electricity-demand-in-victoria-australia). This dataset describes the electricity demand and price situation in Victoria, Australia from 2015 to 2020. The distribution of demand can be seen in Figure EC.1. The descriptive information of the real data is shown in Table 2. The factors that influence daily demand are *price, temperature, solar exposure, school day* and *holiday*. Note that we perform an artificial transformation on the temperature. We define heating degree day (HDD) as $HDD = (T_{min} - 18)^+$, and cooling heating degree day (CDD) as $CDD = (15 - T_{max})^+$, where T_{max} and T_{min} are the highest and lowest centigrade temperatures in one day. This transformation can better reflect the relationship between temperature and electricity demand. The demand is sensitive to price, but it also depend on other features such as temperature and holiday. In our work, we consider the temperature, solar, rainfall, school_day and holiday factors. Note that the scales of features are different, so we standardize the feature to $[0, 1]$ when processing the data. We use Euclidean metric to measure the distance between samples.

Simulation data In terms of simulation data, Note that we refer the demand model to Lin et al. (2022). The demand distribution under $p = 20$ is shown in Figure EC.2. We observe that the distribution is skew and long tail, thus hard to predict by simple models such as linear regression.

EC.3.2 Step size comparison

The step size of CGD algorithm adopted in the numerical experiment part is the Armijo step size with $\sigma = 0$. In Section 3.2, we have analyzed the difference on convergence between the diminishing step size and Armijo step size. In this section, we will evaluate the difference by experiment.

Figure EC.1 Demand distribution for real data

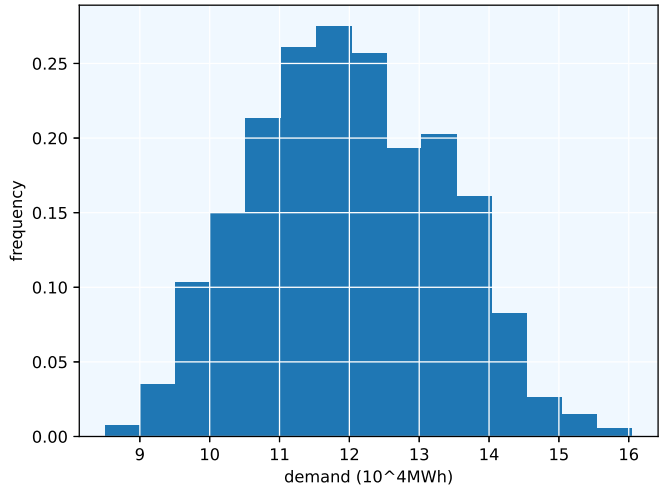
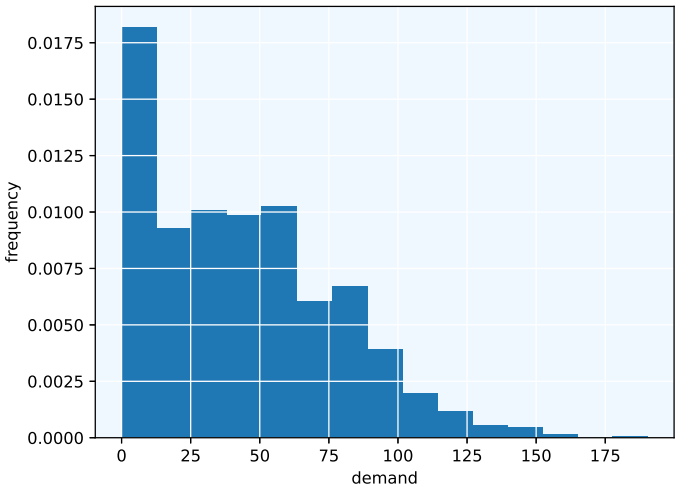


Figure EC.2 Demand distribution for simulated data



We first compare two kinds of step size in the simulated dataset. We set the step size constant $C = 0.05$ in diminishing step size approach. The realized profit and optimality gap are shown in Table EC.1. We can find that the diminishing step size performs worse than Armijo step size in this case. We believe the reason is that the assumptions for the convergence under diminishing step size are usually too strong. The value range Ω in Assumption 1 and L_4, L_5 constant in Proposition 3 maybe large in practice, causing a bad convergence performance. Moreover, we find that although any convergent subsequence converge to the local maximum according to Proposition 3, the diminishing step size cannot stop at the local maximum automatically, which indicates that the diminishing step size may not lead to any convergence subsequence. Therefore, the diminishing step size need a careful selection on the step size constant C and stop criteria.

Table EC.1 Realized profit of Armijo step size and Diminishing step size

Method	kNN	kernel	CART	RF
Armijo	674.37	698.20	689.94	584.07
Diminishing	524.05	512.858	560.0039	388.8681

We also evaluate the effect of hyperparameter σ on CGD algorithm. Table EC.2 reports the optimality gap and iteration number for different constant σ under kernel regression . Figure EC.3 plots the supplementary result in terms of σ and optimality gap. We observe the performance is stable when $\alpha_0 \in (0.01, 0.1)$, and when $\sigma \leq 0.2$. The increment of both α_0 and σ can reduce the iteration numbers, thus accelerate the solution. But when α_0 is larger than 0.5, the optimality gap may become larger. Larger σ can block the update of solution and may cause the algorithm to stop before reaching the convergence.

Table EC.2 Performance comparison among different initial step sizes and σ of Armijo step size

(α_0, σ)	profit	optimality gap	iterations	(α_0, σ)	profit	optimality gap	iterations
(0.01, 0)	695.89	0.97%	90	(0.5, 0)	672.02	4.37%	2
(0.01, 0.1)	688.28	2.05%	74	(0.5, 0.1)	691.25	1.63%	2
(0.01, 0.2)	659.48	6.15%	62	(0.5, 0.2)	667.36	5.03%	2
(0.01, 0.5)	475.37	32.35%	28	(0.5, 0.5)	562.69	19.92%	1
(0.01, 0.9)	300.00	57.31%	0	(0.5, 0.9)	300.00	57.31%	0
(0.05, 0)	698.20	0.64%	18	(1, 0)	653.18	7.05%	1
(0.05, 0.1)	688.84	1.97%	15	(1, 0.1)	653.18	7.05%	1
(0.05, 0.2)	662.67	5.70%	13	(1, 0.2)	653.18	7.05%	1
(0.05, 0.5)	478.73	31.87%	6	(1, 0.5)	569.02	19.02%	1
(0.05, 0.9)	300.00	57.31%	0	(1, 0.9)	300.00	57.31%	0
(0.1, 0)	696.72	0.85%	9				
(0.1, 0.1)	689.42	1.89%	8				
(0.1, 0.2)	659.40	6.16%	7				
(0.1, 0.5)	491.16	30.10%	3				
(0.1, 0.9)	300.00	57.31%	0				

Figure EC.3 Performance comparison among different initial step sizes and σ of Armijo step size