Les Cahiers du GERAD ISSN: 0711–2440

A survey of contextual optimization methods for decision making under uncertainty

U. Sadana, A. Chenreddy, E. Delage, A. Forel, E. Frejinger, T. Vidal

G–2023–22

June 2023

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2023
– Bibliothèque et Archives Canada, 2023

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2023
– Library and Archives Canada, 2023

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

A survey of contextual optimization methods for decision making under uncertainty

Utsav Sadana [a]

Abhilash Chenreddy[b]

Erick Delage [b]

Alexandre Forel [c]

Emma Frejinger [d]

Thibaut Vidal [c]

[a] *GERAD & Desautels Faculty of Management, McGill University, Montréal (Qc), Canada*

[b] *GERAD & Department of Decision Sciences, HEC Montréal, Montréal (Qc), Canada*

[c] *CIRRELT & Department of Mathematical and Industrial Engineering, Polytechnique Montréal, Montréal (Qc), Canada*

[d] *CIRRELT & Department of Computer Science and Operations Research, Université de Montréal, Montréal (Qc), Canada*

utsav.sadana@mail.mcgill.ca
abhilash.chenreddy@hec.ca
erick.delage@hec.ca
alexandre.forel@polymtl.ca
emma.frejinger@umontreal.ca
thibaut.vidal@polymtl.ca

June 2023
Les Cahiers du GERAD
G–2023–22

Abstract : Recently there has been a surge of interest in operations research (OR) and the machine learning (ML) community in combining prediction algorithms and optimization techniques to solve decision-making problems in the face of uncertainty. This gave rise to the field of contextual optimization, under which data-driven procedures are developed to prescribe actions to the decision-maker that make the best use of the most recently updated information. A large variety of models and methods have been presented in both OR and ML literature under a variety of names, including data-driven optimization, prescriptive optimization, predictive stochastic programming, policy optimization, (smart) predict/estimate-then-optimize, decision-focused learning, (task-based) end-to-end learning/forecasting/optimization, etc. Focusing on single and two-stage stochastic programming problems, this review article identifies three main frameworks for learning policies from data and discusses their strengths and limitations. We present the existing models and methods under a uniform notation and terminology and classify them according to the three main frameworks identified. Our objective with this survey is to both strengthen the general understanding of this active field of research and stimulate further theoretical and algorithmic advancements in integrating ML and stochastic programming.

Keywords : Contextual optimization, machine learning, policy optimization, stochastic programming, predict-then-optimize, decision-focused learning, end-to-end learning, implicit differentiation, unrolling

1 Introduction

This article surveys the literature on single and two-stage contextual optimization. In contextual optimization, a decision maker faces a decision-making problem with uncertainty where the distribution of uncertain parameters that affect the objective and the constraints is unknown, although correlated side information (covariates or features) can be exploited. The usefulness of side information in inferring relevant statistics of uncertain parameters and, thereby, in decision-making is evident in many different fields. For example, weather and time of day can help resolve uncertainty about congestion on a road network and aid in finding the shortest path traversing a city. In portfolio optimization, stock returns may depend on historical prices and sentiments posted on Twitter (Xu and Cohen 2018). Harnessing this information can allow decision makers to build a less risky portfolio. Similarly, a retailer facing uncertain demand for summer products can infer whether the demand will be low or high depending on the forecasted weather conditions (Martínez-de Albeniz and Belkaid 2021).

Traditional stochastic optimization models ignore contextual information and use unconditional distributions of the uncertain parameters to make a decision (Birge and Louveaux 2011). Such a decision may be suboptimal (Ban and Rudin 2019) and, in some cases, even at the risk of being infeasible (Rahimian and Pagnoncelli 2022). The availability of data and huge computational power combined with advancements in machine learning (ML) and optimization techniques have resulted in a shift of paradigm to contextual optimization (Mišić and Perakis 2020).

Making prescriptions using the side information requires a decision rule that maps the observed context to an action. We identify three different paradigms for learning this mapping.

Decision rule optimization: This approach was introduced to the operation research community in Ban and Rudin (2019) under the name of big data, although a similar idea was already common practice in reinforcement learning under the name of policy gradient methods (see Sutton et al. 1999, and literature that followed). It consists in employing a parameterized mapping as the decision rule and in identifying the parameter that achieves the best empirical performance based on the available data. The decision rule can be formed as a linear combination of feature functions of the covariates or even using a deep neural network (DNN). When the data available is limited, some form of regularization might also be needed.

Sequential learning and optimization (SLO): Bertsimas and Kallus (2020) appears to be the first to have formalized this two-stage procedure (also referred to as predict/estimate-then-optimize or prescriptive optimization/stochastic programming) that first uses a trained model to predict a conditional distribution for the uncertain parameters given the covariates, and then solves an associated conditional stochastic optimization (CSO) problem to obtain the optimal action. This procedure can be robustified to reduce post-decision disappointment (Smith and Winkler 2006) caused by model overfitting or misspecification by using proper regularization at training time or by adapting the CSO problem formulation.

Integrated learning and optimization (ILO): In the spirit of identifying the best decision rule, one might question in SLO the need for high precision predictors when one is instead mostly interested in the quality of the resulting prescribed action. This idea motivates an integrated version of learning and optimization that searches for the predictive model that guides the CSO problem toward the best-performing actions. The ILO paradigm appears as early as in Bengio (1997) and has seen a resurgence recently in active streams of literature on smart predict-then-optimize, decision-focused learning, and (task-based) end-to-end learning/forecasting/optimization (Donti et al. 2017, Stratigakos et al. 2022, Wahdany et al. 2023).

The outline of the survey goes as follows. Section 2 rigorously defines the three frameworks for learning mappings from data to decisions: decision rule optimization, SLO, and ILO. Section 3 reviews the literature on decision rule optimization with linear and non-linear decision rules. Section 4 focuses on SLO, including the models that lead to robust decisions, and Section 5 describes the models based on the ILO framework and the algorithms used to train them. Section 6 provides an overview of future research directions to pursue both from a theoretical and applications perspective.

We note that there are other surveys and tutorials in the literature that are complementary to ours. Mišić and Perakis (2020) survey the applications of the SLO framework to operations management problems in the supply chain, revenue management, and healthcare. Kotary et al. (2021) provide a comprehensive survey of the literature proposing ML methods to accelerate the resolution of constrained optimization models (see also Bengio et al. 2021). It also reviews some of the older literature on ILO applied to "expected value-based models" (see Definition 1). Qi and Shen (2022) is a tutorial that mainly focuses on the application of ILO to expected value-based models with very limited discussions on more general approaches. It summarizes the most popular methods and some of their theoretical guarantees. In contrast, our survey of ILO goes beyond the expected value-based decision models and reflects better the more modern literature by casting the contextual decision problem as a CSO problem and presenting a comprehensive overview of the current state of this rapidly progressing field of research. We establish links between approaches that minimize regret (Elmachtoub and Grigas 2022),(task-based) end-to-end learning (Donti et al. 2017) and imitation-based models (Kong et al. 2022). Further, we create a taxonomy based on the training procedure for a general ILO framework encompassing recent theoretical and algorithmic progresses in designing differentiable surrogates and optimizers and improving training procedures based on unrolling and implicit differentiation.

2 Contextual optimization: An overview

The contextual optimization paradigm seeks a decision (i.e., an action) $\boldsymbol{z}$ in a feasible set $\mathcal{Z} \subset \mathbb{R}^{n_z}$ that minimizes a cost function $c(\boldsymbol{z};\boldsymbol{y})$ with uncertain parameters $\boldsymbol{y} \in \mathcal{Y} \in \mathbb{R}^{n_y}$. The uncertain parameters are unknown when making the decision. However, a vector of relevant features (covariates) $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$, which is correlated with the uncertain parameters $\boldsymbol{y}$, is revealed before having to choose $\boldsymbol{z}$. The joint distribution of the features in $\mathcal{X}$ and uncertain parameters in $\mathcal{Y}$ is denoted by $\mathbb{P}$.

2.1 Contextual problem and policy

In general, uncertainty can appear in the objectives and constraints of the problem. In the main sections of this paper, we focus on problems with uncertain objectives and consider that the decision maker is risk-neutral. We broaden the discussion to risk-averse settings and uncertain constraints in Section 6.

The CSO problem. Given a context described by a vector of features $\boldsymbol{x}$ and the joint distribution $\mathbb{P}$ of the features $\boldsymbol{x}$ and uncertain parameter $\boldsymbol{y}$, a risk-neutral decision maker is interested in finding an optimal action $\boldsymbol{z}^*(\boldsymbol{x}) \in \mathcal{Z}$ that minimizes the expected costs conditioned on the covariate $\boldsymbol{x}$. Formally, the optimal action is a solution to the conditional stochastic optimization problem given by:

$$\text{(CSO)} \quad \boldsymbol{z}^*(\boldsymbol{x}) \in \operatorname*{argmin}_{\boldsymbol{z}\in\mathcal{Z}} \mathbb{E}_{\mathbb{P}(\boldsymbol{y}|\boldsymbol{x})}\left[c\left(\boldsymbol{z},\boldsymbol{y}\right)\right], \tag{1}$$

where $\mathbb{P}(\boldsymbol{y}|\boldsymbol{x})$ denotes the conditional distribution of $\boldsymbol{y}$ given the covariate $\boldsymbol{x}$ and it is assumed that a minimizer exists. For instance, when $\mathcal{Z}$ is compact, $\mathbb{P}(\boldsymbol{y}|\boldsymbol{x})$ has bounded support and $c(\boldsymbol{z},\boldsymbol{y})$ is continuous in $\boldsymbol{z}$ almost surely (see Van Parys et al. 2021, for more details).

Problem (1) can equivalently be written in a compact form using the expected cost operator $h(\cdot,\cdot)$ that receives a decision as a first argument and a distribution as a second argument:

$$\boldsymbol{z}^*(\boldsymbol{x}) \in \operatorname*{argmin}_{\boldsymbol{z}\in\mathcal{Z}} h(\boldsymbol{z},\mathbb{P}(\boldsymbol{y}|\boldsymbol{x})) := \mathbb{E}_{\mathbb{P}(\boldsymbol{y}|\boldsymbol{x})}\left[c\left(\boldsymbol{z},\boldsymbol{y}\right)\right]. \tag{2}$$

Optimal policy. In general, the decision maker repeatedly solves CSO problems in many different contexts. Hence, the decision maker is interested in finding the policy $\pi \in \Pi$ that provides the lowest long-term expected cost, that is:

$$\pi^* \in \operatorname*{argmin}_{\pi\in\Pi} \mathbb{E}_{\mathbb{P}}\big[c(\pi(\boldsymbol{x}),\boldsymbol{y})\big] = \operatorname*{argmin}_{\pi\in\Pi} \mathbb{E}_{\mathbb{P}}\big[h(\pi(\boldsymbol{x}),\mathbb{P}(\boldsymbol{y}|\boldsymbol{x}))\big]. \tag{3}$$

Note that the optimal policy does not need to be obtained explicitly in a closed form. Indeed, based on the interchangeability property (see Theorem 14.60 of Rockafellar and Wets 2009), solving the CSO Problem (1) in any context $\boldsymbol{x}$ naturally identifies an optimal policy:

$$\bar{\pi}(\boldsymbol{x}) \in \operatorname*{argmin}_{\boldsymbol{z}\in\mathcal{Z}} h(\boldsymbol{z},\mathbb{P}(\boldsymbol{y}|\boldsymbol{x})) \text{ a.s.} \Leftrightarrow \mathbb{E}_{\mathbb{P}}\big[h(\bar{\pi}(\boldsymbol{x}),\mathbb{P}(\boldsymbol{y}|\boldsymbol{x}))\big] = \min_{\pi\in\Pi} \mathbb{E}_{\mathbb{P}}\big[h(\pi(\boldsymbol{x}),\mathbb{P}(\boldsymbol{y}|\boldsymbol{x}))\big],$$

assuming that a minimizer of $h(\boldsymbol{z},\mathbb{P}(\boldsymbol{y}|\boldsymbol{x}))$ exists almost surely. Thus, the two optimal policies π^* and $\boldsymbol{z}^*(\cdot)$ coincide.

2.2 Mapping context to decisions: Three data-driven frameworks

Unfortunately, the joint distribution $\mathbb{P}$ is generally unknown. Instead, the decision maker possesses historical data $\mathcal{D}_N = \{(\boldsymbol{x}_i,\boldsymbol{y}_i)\}_{i=1}^N$ that is assumed to be made of independent and identically distributed realizations of $(\boldsymbol{x},\boldsymbol{y}) \in \mathcal{X}\times\mathcal{Y}$. Using this data, the decision maker aims to find a policy that approximates well the optimal policy given by (3). Many approaches have been proposed to find effective approximate policies. Most of them can be classified into the three main frameworks that we introduce below: (i) decision rule optimization, (ii) sequential learning and optimization, and (iii) integrated learning and optimization.

2.2.1 Decision rule optimization.

In this framework, the decision policy is assumed to belong to a hypothesis class $\Pi^{\boldsymbol{\theta}} := \{\pi_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}\in\boldsymbol{\theta}} \subseteq \Pi$ that contains a family of parametric policies $\pi_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{Z}$ (e.g., linear functions or decision trees). The parameterized policy $\pi_{\boldsymbol{\theta}}$ maps directly any context $\boldsymbol{x}$ to a decision $\pi_{\boldsymbol{\theta}}(\boldsymbol{x})$ and will be referred to as a decision rule.

Denote by $\hat{\mathbb{P}}_N$ the empirical distribution of $(\boldsymbol{x},\boldsymbol{y})$ given historical data $\mathcal{D}_N$. One can identify the "best" policy in $\Pi^{\boldsymbol{\theta}}$ by solving the following empirical risk minimization (ERM) problem:

$$\text{(ERM)} \quad \boldsymbol{\theta}^* \in \operatorname*{argmin}_{\boldsymbol{\theta}} H(\pi_{\boldsymbol{\theta}},\hat{\mathbb{P}}_N) := \mathbb{E}_{\hat{\mathbb{P}}_N}\big[c(\pi_{\boldsymbol{\theta}}(\boldsymbol{x}),\boldsymbol{y})\big]. \tag{4}$$

In simple terms, Problem (4) finds the policy $\pi_{\boldsymbol{\theta}^*} \in \Pi^{\boldsymbol{\theta}}$ that minimizes the expected costs over the training data. Notice that there are two approximations of Problem (3) made by Problem (4). First, the policy is restricted to a hypothesis class that may not contain the true optimal policy. Second, the long-term expected costs are calculated over the empirical distribution $\hat{\mathbb{P}}_N$ rather than the true distribution $\mathbb{P}$. This decision pipeline is shown in Figure 1. Furthermore, Problem (4) focuses its policy optimization efforts on the overall performance (averaged over different contexts) and disregards the question of making the policy achieve a good performance uniformly from one context to another.

2.2.2 Learning and optimization.

The second and third frameworks combine a predictive component and an optimization component. The predictive component is a general model $f_{\boldsymbol{\theta}}$, parameterized by $\boldsymbol{\theta}$, whose role is to provide the input of the optimization component. In any context $\boldsymbol{x}$, the intermediate input $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ can be interpreted as a predicted distribution that approximates the true conditional distribution $\mathbb{P}(\boldsymbol{y}|\boldsymbol{x})$. The predictive component is typically learned from historical data.

At decision time, a learning and optimization decision pipeline (see Figure 2) solves the CSO problem under $f_{\boldsymbol{\theta}}(\boldsymbol{x})$, namely:

$$\boldsymbol{z}^*(\boldsymbol{x}, f_{\boldsymbol{\theta}}) \in \operatorname*{argmin}_{\boldsymbol{z}\in\mathcal{Z}} h(\boldsymbol{z}, f_{\boldsymbol{\theta}}(\boldsymbol{x})). \tag{5}$$

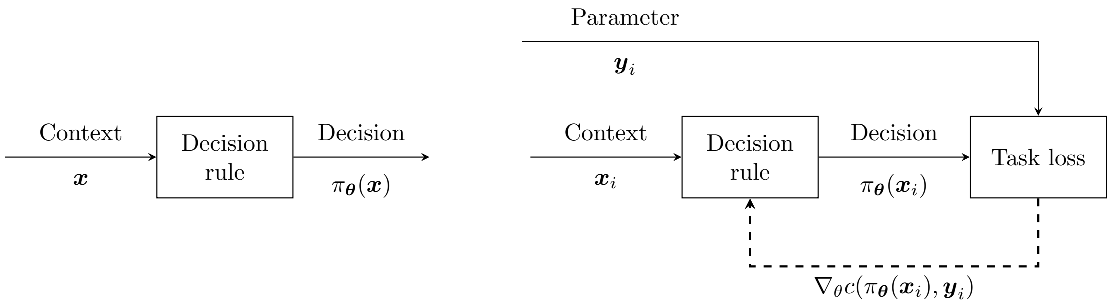

Figure 1: Decision and training pipelines based on the decision rule paradigm: (left) the decision pipeline and (right) the training pipeline for a given training example $(\boldsymbol{x}_i, \boldsymbol{y}_i)$.

The solution of Problem (5) minimizes the expected cost with respect to the predicted distribution $f_{\boldsymbol{\theta}}(\boldsymbol{x})$. Notice that the only approximation between Problem (5) and the true CSO problem in (2) lies in $f_{\boldsymbol{\theta}}$ being an approximation of $\mathbb{P}(\boldsymbol{y}|\boldsymbol{x})$. Since the predicted distribution changes with the context $\boldsymbol{x}$, this pipeline also provides a policy. In fact, if the predictive component were able to perfectly predict the true conditional distribution $\mathbb{P}(\boldsymbol{y}|\boldsymbol{x})$ for any $\boldsymbol{x}$, the pipeline would recover the optimal policy π^* given in (3).

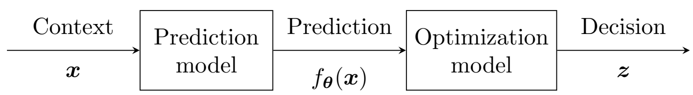

Figure 2: Decision pipeline for learning and optimization.

We now detail the second and third frameworks to address contextual optimization: SLO and ILO. They differ in the way the predictor $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ is trained using the historical data.

Sequential learning and optimization. In this framework, the contextual predictor is obtained by minimizing an estimation error, ρ, between the conditional distribution given by $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ and the true conditional distribution of $\boldsymbol{y}$ given $\boldsymbol{x}$. Training the contextual predictor implies solving the following estimation problem:

$$\min_{\boldsymbol{\theta}} \rho(f_{\boldsymbol{\theta}}, \hat{\mathbb{P}}_N) + \Omega(\boldsymbol{\theta}) \text{ with } \rho(f_{\boldsymbol{\theta}}, \hat{\mathbb{P}}_N) := \mathbb{E}_{\hat{\mathbb{P}}_N}[\mathfrak{D}(f_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y})], \tag{6}$$

where $\mathfrak{D}$ is a divergence function, e.g., negative log-likelihood (NLL) and the regularization term $\Omega(\boldsymbol{\theta})$ controls the complexity of $f_{\boldsymbol{\theta}}$. The SLO training pipeline is shown in Figure 3.

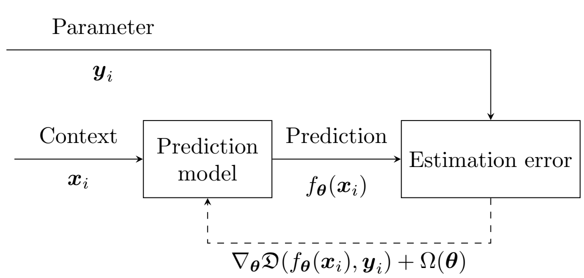

Figure 3: SLO training pipeline for a given training example.

Definition 1 (Expected value-based models)**.** When the cost function $c(\boldsymbol{x}, \boldsymbol{y})$ of the decision model is linear in $\boldsymbol{y}$, the problem of estimating a conditional distribution reduces to finding the expected value of the uncertain parameter given the covariates since $h(\boldsymbol{z}, \mathbb{P}(\boldsymbol{y}|\boldsymbol{x})) = \mathbb{E}_{\mathbb{P}(\boldsymbol{y}|\boldsymbol{x})}[c(\boldsymbol{z}, \boldsymbol{y})] = c(\boldsymbol{z}, \mathbb{E}_{\mathbb{P}(\boldsymbol{y}|\boldsymbol{x})}[\boldsymbol{y}])$. Training the contextual predictor, therefore, reduces to a mean regression problem over a parameterized function $g_{\boldsymbol{\theta}}(\boldsymbol{x})$. Specifically,

$$\min_{\boldsymbol{\theta}} \rho(g_{\boldsymbol{\theta}}, \hat{\mathbb{P}}_N) + \Omega(\boldsymbol{\theta}) \text{ with } \rho(g_{\boldsymbol{\theta}}, \hat{\mathbb{P}}_N) := \mathbb{E}_{\hat{\mathbb{P}}_N}\left[d(g_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y})\right], \tag{7}$$

for some distance metric d, usually the sum of squared errors. While the sum of squared error is known to asymptotically retrieve $g^*_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathbb{E}_{\mathbb{P}(\boldsymbol{y}|\boldsymbol{x})}[\boldsymbol{y}]$ under standard conditions, other distance metric or more general loss functions can also be used (Hastie et al. 2009). For any new context $\boldsymbol{x}$, a decision is obtained using:

$$z^*(\boldsymbol{x}, g_{\boldsymbol{\theta}}) \in \operatorname*{argmin}_{\boldsymbol{z}\in\mathcal{Z}} h(\boldsymbol{z}, \delta_{g_{\boldsymbol{\theta}}(\boldsymbol{x})}) = \operatorname*{argmin}_{\boldsymbol{z}\in\mathcal{Z}} c(\boldsymbol{z}, g_{\boldsymbol{\theta}}(\boldsymbol{x})), \tag{8}$$

with $\delta_{\boldsymbol{y}}$ being the Dirac distribution putting all its mass at $\boldsymbol{y}$. In the remainder of this survey, we refer to these approaches as **expected value-based models**, while the more general models that prescribe using a conditional distribution estimator (i.e. $z^*(\boldsymbol{x}, f_{\boldsymbol{\theta}})$) will be referred as a **conditional distribution-based models** when it is not clear from the context.

Integrated learning and optimization. Sequential approaches ignore the mismatch between the prediction divergence $\mathfrak{D}$ and the cost function $c(\boldsymbol{x}, \boldsymbol{y})$. Depending on the context, a small prediction error about $\mathbb{P}(\boldsymbol{y}|\boldsymbol{x})$ may have a large impact on the prescription. In integrated learning, the goal is to maximize the prescriptive performance of the induced policy. That is, we want to train the predictive component to minimize the task loss (i.e., the downstream costs incurred by the decision) as stated in (3). The prescriptive performance may guide the estimation procedure toward an inaccurate prediction that produces a nearly-optimal decision. This is illustrated in Figure 4.

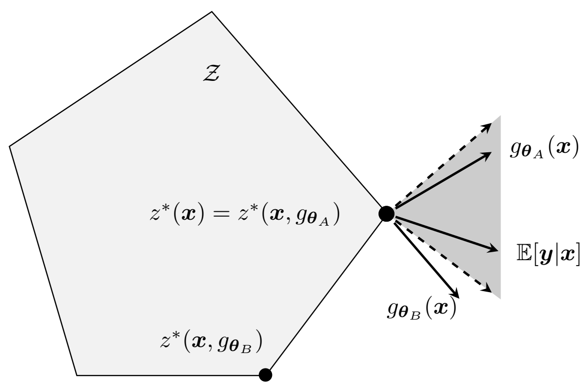

Figure 4: Predicting $g_{\boldsymbol{\theta}_A}(\boldsymbol{x})$ results in the optimal decision $z^*(\boldsymbol{x}, g_{\boldsymbol{\theta}_A}) = z^*(\boldsymbol{x})$ whereas a small error resulting from predicting $g_{\boldsymbol{\theta}_B}(\boldsymbol{x})$ leads to a suboptimal decision $z^*(\boldsymbol{x}, g_{\boldsymbol{\theta}_B})$ under $c(\boldsymbol{x}, \boldsymbol{y}) := -\boldsymbol{y}^\top \boldsymbol{x}$, i.e., $h(\boldsymbol{z}, \mathbb{P}(\boldsymbol{y}|\boldsymbol{x})) = -\mathbb{E}[\boldsymbol{y}|\boldsymbol{x}]^\top \boldsymbol{z}$ (adapted from Elmachtoub and Grigas 2022).

Finding the best parameterization of a contextual predictor that minimizes the downstream expected costs of the CSO solution can be formulated as the following problem:

$$\min_{\boldsymbol{\theta}} H(\boldsymbol{z}^*(\cdot, f_{\boldsymbol{\theta}}), \hat{\mathbb{P}}_N) = \min_{\boldsymbol{\theta}} \mathbb{E}_{\hat{\mathbb{P}}_N}\left[c(\boldsymbol{z}^*(\boldsymbol{x}, f_{\boldsymbol{\theta}}), \boldsymbol{y})\right]. \tag{9}$$

The objective function in (9) minimizes the average cost of the policy over the empirical distribution. The policy induced by this training problem is thus optimal with respect to the predicted distribution for each historical sample and minimizes the average historical costs over the whole training data. Figure 5 describes how the downstream cost is propagated by the predictive model during training. This training procedure necessarily comes at the price of heavier computations because an optimization model needs to be solved for each data point, and differentiation needs to be applied through an argmin operation.

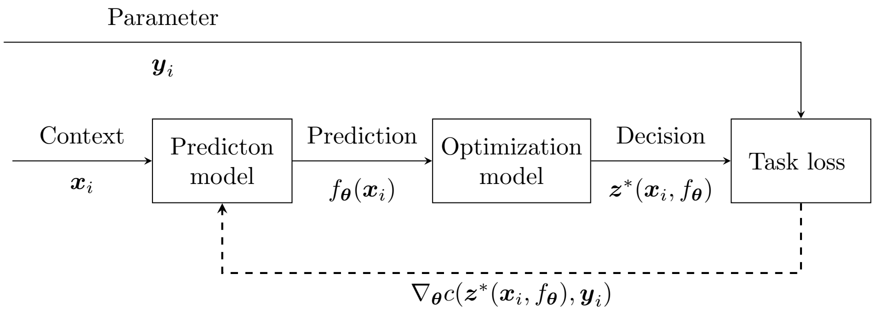

Figure 5: ILO training pipeline for a given training example.

2.3 Summary

This section presents the main pipelines proposed in recent years to address contextual optimization. Although these pipelines all include a learning component, they differ significantly in their specific structures and training procedures. Overall, there are several design choices that the decision maker should make when tackling contextual optimization: (i) the architecture of the pipeline, i.e., whether it includes a single predictive component or whether it combines learning and optimization, (ii) the class of the predictive model (e.g., linear, neural network or random forest) and its hyperparameters, (iii) the type of loss function used during training, i.e., minimizing the estimation error or the downstream cost of the policy, which defines whether an approach belongs to the sequential or integrated paradigm. Each design choice has its own inductive bias and may imply specific methodological challenges, especially for ILO. In general, it is a priori unclear what combination of choices will lead to the best performance; therefore, pipelines have to be evaluated experimentally.

In the following sections, we survey the recent literature corresponding to the three main frameworks for contextual optimization using the notation introduced so far, which is summarized in Table 1. A list of abbreviations used in this survey is given in Appendix 1.2.

Table 1: Notation: distributions, variables, and operators.

	Domain	Description
$\mathbb{P}$	$\mathcal{M}(\mathcal{X} \times \mathcal{Y})$	True (unknown) joint distribution of $(\boldsymbol{x}, \boldsymbol{y})$
$\hat{\mathbb{P}}_N$	$\mathcal{M}(\mathcal{X} \times \mathcal{Y})$	Joint empirical distribution of $(\boldsymbol{x}, \boldsymbol{y})$
δ_y	$\mathcal{M}(\mathcal{Y})$	Dirac distribution that puts all of its weight on $\boldsymbol{y}$
$\boldsymbol{x}$	$\mathcal{X} \subseteq \mathbb{R}^{n_x}$	Contextual information
$\boldsymbol{y}$	$Y \subseteq \mathbb{R}^{n_y}$	Uncertain parameters
$\boldsymbol{z}$	$\mathcal{Z} \subseteq \mathbb{R}^{n_z}$	A feasible decision
$\boldsymbol{\theta}$	Θ	Parameters of a prediction model
$c(\boldsymbol{z}, \boldsymbol{y})$	$\mathbb{R}$	Cost of a decision $\boldsymbol{z}$ under $\boldsymbol{y}$
$h(\boldsymbol{z}, \mathbb{Q}_y)$	$\mathbb{R}$	Expected cost of a decision $\boldsymbol{z}$ under $\mathbb{Q}_y$ (a distribution over $\boldsymbol{y}$)
$H(\pi, \mathbb{Q})$	$\mathbb{R}$	Expected cost of a policy π under $\mathbb{Q}$ (a distribution over $(\boldsymbol{x}, \boldsymbol{y})$)
$f_{\boldsymbol{\theta}}(\boldsymbol{x})$	$\mathcal{H}$	Estimate of the conditional distribution of $\boldsymbol{y}$ given $\boldsymbol{x}$
$g_{\boldsymbol{\theta}}(\boldsymbol{x})$	$\mathbb{R}$	Estimate of the conditional expectation of $\boldsymbol{y}$ given $\boldsymbol{x}$
$\pi^*(\boldsymbol{x})$	$\mathcal{Z}$	Optimal solution of CSO under true conditional distribution $\mathbb{P}(\boldsymbol{y}\vert\boldsymbol{x})$
$\pi_{\boldsymbol{\theta}}(\boldsymbol{x})$	$\mathcal{Z}$	Decision prescribed by a policy parameterized by $\boldsymbol{\theta}$ for context $\boldsymbol{x}$
$z^*(\boldsymbol{x})$	$\mathcal{Z}$	Optimal solution to the CSO problem under the true conditional distribution $\mathbb{P}(\boldsymbol{y}\vert\boldsymbol{x})$
$z^*(\boldsymbol{x}, f_{\boldsymbol{\theta}})$	$\mathcal{Z}$	Optimal solution to the CSO problem under the conditional distribution $f_{\boldsymbol{\theta}}(\boldsymbol{x})$
$z^*(\boldsymbol{x}, g_{\boldsymbol{\theta}})$	$\mathcal{Z}$	Optimal solution to the CSO problem under the Dirac distribution $\delta_{g_{\boldsymbol{\theta}}(\boldsymbol{x})}$
$\rho(f_{\boldsymbol{\theta}}, \hat{\mathbb{P}}_N)$	$\mathbb{R}$	Expected prediction error for distribution model $f_{\boldsymbol{\theta}}$ based on empirical distribution in $\hat{\mathbb{P}}_N$
$\rho(g_{\boldsymbol{\theta}}, \hat{\mathbb{P}}_N)$	$\mathbb{R}$	Expected prediction error for point prediction model $g_{\boldsymbol{\theta}}$ based on empirical distribution in $\hat{\mathbb{P}}_N$

3 Decision rule optimization

Decision rules obtained by solving the ERM in Problem (4) minimize the cost of a policy on the task, that is, the downstream optimization problem. Policy-based approaches are especially efficient computationally at decision time since it suffices to evaluate the estimated policy. No optimization problem needs to be solved once the policy is trained. We defined the decision rule approach as employing a parameterized mapping $\pi_{\boldsymbol{\theta}}(\boldsymbol{x})$, e.g., linear policies (Ban and Rudin 2019) or a neural network (Oroojlooyjadid et al. 2020). Since policies obtained using neural networks lack interpretability, linear decision rules are widely used.

3.1 Linear decision rules

Ban and Rudin (2019) show that an approach based on the sample-average approximation (SAA) that disregards side information can lead to inconsistent decisions (i.e., asymptotically suboptimal) for a newsvendor problem. Using linear decision rules (LDRs), they study two variants of the newsvendor problem with and without regularization:

$$\min_{\pi:\pi(\boldsymbol{x})=\boldsymbol{q}^\top\boldsymbol{x}} H(\pi,\hat{\mathbb{P}}_N)+\Omega(\pi)=\min_{\boldsymbol{q}}\frac{1}{N}\sum_{i=1}^{N}u(y_i-\boldsymbol{q}^\top\boldsymbol{x}_i)^+ + o(\boldsymbol{q}^\top\boldsymbol{x}_i-y_i)^+ + \lambda\|\boldsymbol{q}\|_k^2,$$

where o and u denote the per unit backordering (underage) and holding (overage) costs. Ban and Rudin (2019) show that for a linear demand model, the generalization error for the ERM model scales as $\mathrm{O}(\frac{n_x}{\sqrt{N}})$ when there is no regularization and as $\mathrm{O}(\frac{n_x}{\sqrt{N}\lambda})$ with regularization. However, one needs to balance the trade-off between generalization error and bias due to regularization to get the optimal performance from using LDRs. Ban and Rudin (2019) and Bertsimas and Kallus (2020) consider unconstrained problems because it is difficult to ensure the feasibility of policies and maintain computational tractability using the ERM approach. Unfortunately, LDRs may not be asymptotically optimal in general, and the out-of-sample guarantees have only been established for specific classes of policies (e.g., LDRs). To generalize LDRs, one can consider decision rules that are linear in the transformations of the covariate vector (Ban and Rudin 2019). It is also possible to lift the covariate vector to a reproducing kernel Hilbert space (RKHS) (Aronszajn 1950), as seen in the next section.

3.2 RKHS-based decision rules

To obtain decision rules that are more flexible than linear ones with respect to $\boldsymbol{x}$, it is possible to lift the covariate vector to an RKHS in which LDRs might achieve better performance. Let $K:\mathcal{X}\times\mathcal{X}\to\mathbb{R}$ be the symmetric positive definite kernel associated with the chosen RKHS, e.g., the Gaussian kernel $K(\boldsymbol{x}_1,\boldsymbol{x}_2):=\exp(-\|\boldsymbol{x}_1-\boldsymbol{x}_2\|^2/(2\sigma^2))$. Given K, the RKHS $\mathcal{H}_K$ is defined as the closure of a set of functions given below:

$$\Big\{\varphi:\mathcal{X}\to\mathbb{R}\,|\,\exists\, L\in\mathbb{N},\,\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_L\in\mathcal{X},\,\varphi(\boldsymbol{x})=\sum_{l=1}^{L}a_lK(\boldsymbol{v}_l,\boldsymbol{x}),\forall\boldsymbol{x}\in\mathcal{X}\Big\},$$

with the inner product of $\varphi_1(\boldsymbol{x})=\sum_{i=1}^{L_1}a_1^iK(\boldsymbol{v}_1^i,\boldsymbol{x})$ and $\varphi_2(\boldsymbol{x})=\sum_{j=1}^{L_2}a_2^jK(\boldsymbol{v}_2^j,\boldsymbol{x})$ given by:

$$\langle\varphi_1,\varphi_2\rangle=\sum_{i=1}^{L_1}\sum_{j=1}^{L_2}a_1^ia_2^jK(\boldsymbol{v}_1^i,\boldsymbol{v}_2^j).$$

Bertsimas and Koduri (2022) approximate the optimal policy with a linear policy in the RKHS, i.e. $\pi_\varphi(\boldsymbol{x}):=\langle\varphi,K(\boldsymbol{x},\cdot)\rangle$ when $n_z=1$, and show using the representer theorem (see Theorem 9 in Hofmann et al. 2008) that the solution of the following regularized problem:

$$\min_{\varphi\in\mathcal{H}_K}H(\pi_\varphi,\hat{\mathbb{P}}_N)+\lambda\|\varphi\|_2^2,$$

takes the form $\pi^*(\boldsymbol{x}) = \sum_{i=1}^{N} K(\boldsymbol{x}_i, \boldsymbol{x}) a_i^*$. Hence, this reduces the decision rule problem to:

$$\min_{\boldsymbol{a}\in\mathbb{R}^N} H\left(\sum_{i=1}^{N} K(\boldsymbol{x}_i, \cdot) a_i, \hat{\mathbb{P}}_N\right) + \lambda \sum_{i=1}^{N}\sum_{j=1}^{N} K(\boldsymbol{x}_i, \boldsymbol{x}_j) a_i a_j.$$

This can be extended to $n_z > 1$ by employing one RKHS for each z_i.

This RKHS approach appeared earlier in Ban and Rudin (2019) and Bazier-Matte and Delage (2020) who respectively study the data-driven single item newsvendor and single risky asset portfolio problems and establish bounds on the out-of-sample performance. Bertsimas and Koduri (2022) show the asymptotic optimality of RKHS-based policies. Notz and Pibernik (2022) study a two-stage capacity planning problem with multivariate demand and vector-valued capacity decisions for which the underlying demand distribution is difficult to estimate in practice. Similar to Bazier-Matte and Delage (2020), the authors optimize over policies that are linear in the RKHS associated with the Gaussian kernel and identify generalization error bounds. For large dimensional problems, this kernel is shown to have a slow convergence rate, and as a result, the authors propose instead using a data-dependent random forest kernel.

3.3 Non-linear decision rules

Many non-linear decision rule approaches have been experimented with. Zhang and Gao (2017), Huber et al. (2019), and Oroojlooyjadid et al. (2020) study the value of training a DNN to learn the ordering policy of a newsvendor problem. It is well-known that neural networks enjoy the universal approximation property; that is, they can approximate any continuous function arbitrarily well (Cybenko 1989, Lu et al. 2017). For constrained problems, one can use softmax as the final layer to ensure that decisions lie in a simplex, e.g., in a portfolio optimization problem (Zhang et al. 2021). In practice, the output of a neural network might not naturally land in the feasible space $\mathcal{Z}$. To circumvent this issue, Chen et al. (2023) proposed to introduce an application-specific differentiable repair layer that projects the solution back to feasibility. Rychener and Sutter (2023) show that the decision rule obtained by using the stochastic gradient descent (SGD) method to train DNN-based policies approximately minimizes the Bayesian posterior loss.

Exploiting the fact that the optimal solution of a newsvendor problem is a quantile of the demand distribution, Huber et al. (2019) further trains, using quantile regression, an additive ensemble of decision trees to produce the ordering decision. They test these algorithms on a real-world dataset from a large German bakery chain. Bertsimas et al. (2019), Ciocan and Mišić (2022), and Keshavarz (2022) optimize decision tree-based decision rules to address the multi-item newsvendor, treatment planning, and optimal stopping problems, respectively. A tutorial on DNN-based decision rule optimization is given in Shlezinger et al. (2022).

Zhang et al. (2023b) introduce piecewise-affine decision rules (PADRs) and provide non-asymptotic and asymptotic consistency results for unconstrained and constrained problems, respectively. The policy is learned through a stochastic majorization-minimization algorithm. Experiments on a constrained newsvendor problem show that PADRs can outperform the RKHS-based policies.

3.4 Distributionally robust decision rules

Most of the literature on policy learning assumes a parametric form $\Pi^{\boldsymbol{\theta}}$ for the policy. A notable exception is Zhang et al. (2023a), which study a distributionally robust contextual newsvendor problem under type-1 Wasserstein ambiguity set, without assuming an explicit structure on the policy class. The distributionally robust model avoids the degeneracies of ERM for generic Π by defining an optimal "Shapley" extension to the scenario-based optimal policy. Mathematically,

$$\min_{\pi\in\Pi} \sup_{\mathbb{Q}\in\mathcal{M}(\mathcal{X}\times\mathcal{Y})} \{H(\pi,\mathbb{Q}) : \mathcal{W}(\mathbb{Q},\hat{\mathbb{P}}_N) \le r\} \equiv \min_{\pi:\hat{\mathcal{X}}\to\mathcal{Z}} \sup_{\mathbb{Q}\in\mathcal{M}(\hat{\mathcal{X}}\times\mathcal{Y})} \{H(\pi,\mathbb{Q}) : \mathcal{W}(\mathbb{Q},\hat{\mathbb{P}}_N) \le r\},$$

where $\hat{\mathcal{X}} := \cup_{i=1}^{N}\{\boldsymbol{x}_i\}$ and $\mathcal{M}(\hat{\mathcal{X}} \times \mathcal{Y})$ is the set of all distribution supported on $\hat{\mathcal{X}} \times \mathcal{Y}$.

Prior to the work of Zhang et al. (2023a), many have considered distributionally robust versions of the decision rule optimization problem in the non-contextual setting (Yanıkoğlu et al. 2019) while Bertsimas et al. (2023) use LDRs to solve dynamic optimization problems with side information.

Yang et al. (2023) point out that the perturbed distributions in the Wasserstein ambiguity set might have a different conditional information structure than the estimated conditional distribution. They introduce a distributionally robust optimization (DRO) problem with causal transport metric (Backhoff et al. 2017, Lassalle 2018) that places an additional causality constraint on the transport plan compared to the Wasserstein metric. Tractable reformulations of the DRO problem are given under LDRs as well as for one-dimensional convex cost functions. Rychener and Sutter (2023) present a Bayesian interpretation of decision rule optimization using SGD and show that their algorithm provides an unbiased estimate of the worst-case objective function of a DRO problem as long as a uniqueness condition is satisfied. The authors note that the Wasserstein ambiguity set violates this condition and thus use the Kullback-Leibler (KL) divergence (Kullback and Leibler 1951) to train the models.

4 Sequential learning and optimization

In reviewing contextual optimization approaches that are based on SLO, we distinguish two settings: (i) a more traditional setting where the conditional distribution is learned from data and used directly in the optimization model, and (ii) a setting that attempts to produce decisions that are robust to model misspecification. An overview of the methods presented in this section is given in Table 2.

Table 2: Overview of contextual optimization papers in the SLO framework.

	Method			Regularization		Learning model					
	`rCSO`	`wSAA`	`EVB`	Reg. CSO	DRO	General	Linear	Kernel	kNN	DT	RF
Deng and Sen (2022)	✔	✗	✗	✗	✗	✔	✔	✔	✔	✔	✔
Ban et al. (2019)	✔	✗	✗	✗	✗	✗	✔	✗	✗	✗	✗
Kannan et al. (2020)	✔	✗	✗	✗	✗	✔	✔	✔	✔	✔	✔
Hannah et al. (2010)	✗	✔	✗	✗	✗	✗	✗	✔	✗	✗	✗
Bertsimas and Kallus (2020)	✗	✔	✗	✗	✗	✗	✔	✗	✔	✔	✔
Notz and Pibernik (2022)	✗	✔	✗	✗	✗	✗	✗	✔	✗	✔	✗
Ferreira et al. (2016)	✗	✗	✔	✗	✗	✗	✗	✗	✗	✔	✗
Liu et al. (2021)	✗	✗	✔	✗	✗	✗	✔	✗	✗	✔	✗
Lin et al. (2022)	✗	✔	✗	✔	✗	✗	✗	✗	✔	✔	✔
Srivastava et al. (2021)	✗	✔	✗	✔	✗	✗	✗	✔	✗	✗	✗
Bertsimas and Van Parys (2022)	✗	✔	✗	✗	✔	✗	✗	✔	✔	✗	✗
Wang et al. (2021)	✗	✔	✗	✗	✔	✗	✗	✔	✗	✗	✗
Chen and Paschalidis (2019)	✗	✔	✗	✗	✔	✗	✗	✗	✔	✗	✗
Nguyen et al. (2021)	✗	✔	✗	✗	✔	✗	✗	✗	✔	✗	✗
Esteban-Pérez and Morales (2022)	✗	✔	✗	✗	✔	✗	✗	✔	✔	✗	✗
Kannan et al. (2021)	✔	✗	✗	✗	✔	✔	✔	✔	✔	✔	✔
Kannan et al. (2022)	✔	✗	✗	✗	✔	✔	✔	✔	✔	✔	✔
Perakis et al. (2023)	✔	✗	✗	✗	✔	✗	✔	✗	✗	✗	✗
Zhu et al. (2022)	✗	✗	✔	✗	✔	✔	✔	✔	✔	✔	✔

Note: we distinguish between regularized CSO models (Reg. CSO) and DRO-based regularization; an approach is classified as"General" if its learning model is not restricted to specific classes.

4.1 Learning conditional distributions

Most of the recent literature has employed discrete models for $f_{\boldsymbol{\theta}}(\boldsymbol{x})$. This is first motivated from a computational perspective by the fact that the CSO Problem (5) is easier to solve in this setting. In fact, more often than not, the CSO under a continuous distribution needs to be first replaced by its SAA to be solved (Shapiro et al. 2014). From a statistical viewpoint, it can also be difficult to assess the probability of outcomes that are not present in the dataset, thus justifying fixing the support of $\boldsymbol{y}$ to its observed values.

4.1.1 Residual-based distribution.

A first approach (found in Sen and Deng 2017, Kannan et al. 2020, Deng and Sen 2022) is to use the errors of a trained regression model (i.e., its residuals) to construct conditional distributions. Let $g_{\hat{\boldsymbol{\theta}}}$ be a regression model trained to predict the response $\boldsymbol{y}$ from the covariate $\boldsymbol{x}$, thus minimizing an estimation error ρ as in (7). The residual error of sample i is given by $\boldsymbol{\epsilon}_i = \boldsymbol{y}_i - g_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}_i)$. The set of residuals measured on the historical data, $\{\boldsymbol{\epsilon}_i\}_{i=1}^N$, is then used to form the conditional distribution $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^N \delta_{g_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})+\boldsymbol{\epsilon}_i}$. The residual-based CSO (`rCSO`) problem is now given by:

$$(\texttt{rCSO}) \quad \min_{\boldsymbol{z}\in\mathcal{Z}} h(\boldsymbol{z}, \mathbb{P}^{\text{ER}}(\boldsymbol{x})) \text{ with } \mathbb{P}^{\text{ER}}(\boldsymbol{x}) := \frac{1}{N}\sum_{i=1}^N \delta_{\text{proj}_{\mathcal{Y}}(\text{g}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})+\boldsymbol{\epsilon}_\text{i})}. \tag{10}$$

with $\text{proj}_{\mathcal{Y}}$ denoting the orthogonal projection on the support $\mathcal{Y}$. The advantage of residual-based methods is that they can be applied in conjunction with any trained regression model. Ban et al. (2019) and Deng and Sen (2022) build conditional distributions for two-stage and multi-stage CSO problems using the residuals obtained from parametric regression on the historical data.

Notice that, in this approach, the historical data is used twice: to train the regression model $g_{\boldsymbol{\theta}}$, and to measure the residuals $\boldsymbol{\epsilon}_i$. This can lead to an underestimation of the distribution of the residual error. To remove this bias, Kannan et al. (2020) propose a leave-one-out model (also known as *jackknife*). They measure the residuals as $\tilde{\boldsymbol{\epsilon}}_i = \boldsymbol{y}_i - g_{\hat{\boldsymbol{\theta}}_{-i}}(\boldsymbol{x}_i)$, where $\hat{\boldsymbol{\theta}}_{-i}$ is trained using all the historical data except the i-th sample $(\boldsymbol{x}_i, \boldsymbol{y}_i)$. This idea can also be applied to the heteroskedastic case studied in Kannan et al. (2021), where the following conditional distribution is obtained by first estimating the conditional covariance matrix $\hat{Q}(\boldsymbol{x})$ (a positive definite matrix for almost every $\boldsymbol{x}$) and then forming the residuals $\hat{\boldsymbol{\epsilon}}_i = [\hat{Q}(\boldsymbol{x}_i)]^{-1}(\boldsymbol{y}_i - g_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}_i))$:

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) := \frac{1}{N}\sum_{i=1}^N \delta_{\text{proj}_{\mathcal{Y}}(\text{g}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})+\hat{\text{Q}}(\boldsymbol{x})\hat{\boldsymbol{\epsilon}}_\text{i})}.$$

4.1.2 Weight-based distribution.

A typical approach for formulating the CSO problem is to assign weights to the observations of the uncertain parameters in the historical data and solving the weighted SAA problem (`wSAA`) given by (Bertsimas and Kallus 2020):

$$(\texttt{wSAA}) \quad \min_{z\in\mathcal{Z}} h\left(\boldsymbol{z}, \sum_{i=1}^N w_i(\boldsymbol{x})\delta_{\boldsymbol{y}_i}\right). \tag{11}$$

In this case, the conditional distribution $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{i=1}^N w_i(\boldsymbol{x})\delta_{\boldsymbol{y}_i}$ is fully determined by the function used to assign a weight to the historical samples. Different approaches have been proposed to determine the sample weights with ML methods.

Weights based on proximity. Sample weights can be assigned based on the distance between a context $\boldsymbol{x}$ and each historical sample $\boldsymbol{x}_i$. For instance, a k-nearest neighbor (kNN) estimation gives equal weight to the k closest samples in the dataset and zero weight to all the other samples. That is, $w_i^{\text{kNN}}(\boldsymbol{x}) := (1/k)\mathbb{1}[\boldsymbol{x}_i \in \mathcal{N}_k(\boldsymbol{x})]$, where $\mathcal{N}_k(\boldsymbol{x})$ denotes the set of k nearest neighbors of $\boldsymbol{x}$ and $\mathbb{1}[\cdot]$ is the indicator function. Even though it may appear simple, this non-parametric approach benefits from asymptotic consistency guarantees on its prescriptive performance. Another method to determine sample weights is to use kernel density estimators (Hannah et al. 2010, Srivastava et al. 2021, Ban and

Rudin 2019). The Nadaraya-Watson (NW) kernel estimator (Watson 1964, Nadaraya 1964) employs a weight function:

$$w_i^{\mathrm{KDE}}(\boldsymbol{x}) := \frac{K\left((\boldsymbol{x}-\boldsymbol{x}_i)/\boldsymbol{\theta}\right)}{\sum_{j=1}^{N} K\left((\boldsymbol{x}-\boldsymbol{x}_j)/\boldsymbol{\theta}\right)}$$

where K is a kernel function and $\boldsymbol{\theta}$ denotes its bandwidth parameter. Different kernel functions can be used, e.g., the Gaussian kernel defined as $K(\boldsymbol{\Delta}) \propto \exp(-\|\boldsymbol{\Delta}\|^2)$. Hannah et al. (2010) also use Bayesian approach that exploits the Dirichlet process mixture to assign sample weights.

Weights based on random forest. Weights can also be designed based on random forests (Bertsimas and Kallus 2020). In its simplest setting, the weight function of a decision tree is given by:

$$w_i^t(\boldsymbol{x}) := \frac{\mathbb{1}\left[\mathcal{R}_t(\boldsymbol{x}) = \mathcal{R}_t(\boldsymbol{x}_i)\right]}{\sum_{j=1}^{N} \mathbb{1}\left[\mathcal{R}_t(\boldsymbol{x}) = \mathcal{R}_t(\boldsymbol{x}_j)\right]}$$

where $\mathcal{R}_t(\boldsymbol{x})$ denotes the terminal node of tree t that contains covariate $\boldsymbol{x}$. Thus, a decision tree assigns equal weights to all the historical samples that end in the same leaf node as $\boldsymbol{x}$. The random forest weight function generalizes this idea over many random decision trees. Its weight function is defined as:

$$w_i^{\mathrm{RF}}(\boldsymbol{x}) := \frac{1}{T}\sum_{t=1}^{T} w_i^t(\boldsymbol{x}),$$

where w_i^t is the weight function of tree t. Random forests are typically trained in order to perform an inference task, e.g. regression, or classification, but can also be used and interpreted as non-parametric conditional density estimators.

Bertsimas and Kallus (2020) provide conditions for the asymptotic optimality (see Definition A1 in Appendix 1.1) and consistency (see Definition A2 in Appendix 1.1) of prescriptions obtained by solving Problem (11) with the weights functions given by kNN, NW kernel density estimation and local linear regression.

4.1.3 Expected value-based models.

As described in Definition 1, when the cost function is linear, the training pipeline of SLO reduces to conditional mean estimation. For instance, Ferreira et al. (2016) train regression trees to forecast daily expected sales for different product categories in an inventory and pricing problem for an online retailer. Alternatively, one may attempt to approximate the conditional density $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ using a point prediction $g_{\boldsymbol{\theta}}(\boldsymbol{x})$. For example, Liu et al. (2021) study a last-mile delivery problem, where customer orders are assigned to drivers, and replace the conditional distribution of the stochastic travel time with a point predictor (e.g. a linear regression or decision tree) that accounts for the number of stops, total distance of the trip, etc.

4.2 Regularization and distributionally robust optimization

While non-parametric conditional density estimation methods benefit from asymptotic consistency (Bertsimas and Kallus 2020, Notz and Pibernik 2022), they are known to produce overly optimistic policies when the size of the covariate vector is large (see discussions in Bertsimas and Van Parys 2022). To circumvent this issue, authors have proposed to either regularize the CSO problem (Srivastava et al. 2021, Lin et al. 2022) or to cast it as a DRO problem. In the latter case, one attempts to minimize the worst-case expected cost over the set of distributions $\mathcal{B}_r(f_{\boldsymbol{\theta}}(\boldsymbol{x}))$ that lie at a distance r from the estimated distribution $f_{\boldsymbol{\theta}}(\boldsymbol{x})$:

$$\min_{z\in\mathcal{Z}} \sup_{\mathbb{Q}_y \in \mathcal{B}_r(f_{\boldsymbol{\theta}}(\boldsymbol{x}))} h(\boldsymbol{z}, \mathbb{Q}_y).$$

Bertsimas and Van Parys (2022) generate bootstrap data from the training set and use it as a proxy for the "out-of-sample disappointment" of a decision $\boldsymbol{z}$ resulting from the out-of-sample cost exceeding the budget given by $\sup_{\mathbb{Q}_y \in \mathcal{B}_r(f_{\boldsymbol{\theta}}(\boldsymbol{x}))} h(\boldsymbol{z}, \mathbb{Q}_y)$. They show that for the NW kernel density estimator and nearest neighbor estimator, the DRO, under a range of ambiguity sets, can be reformulated as a convex optimization problem. Using KL divergence to measure the distance between the probability distributions, they obtain guarantees ("bootstrap robustness") with respect to the estimate-then-optimize model taking bootstrap data as a proxy for out-of-sample data. Taking the center of Wasserstein ambiguity set (see Kantorovich and Rubinshtein 1958) to be NW kernel estimator, Wang et al. (2021) show that the distributionally robust newsvendor and conditional value at risk (CVaR) portfolio optimization problems can be reformulated as convex programs where the type-p Wasserstein distance (earth mover's distance) between distributions $\mathbb{P}_1$ and $\mathbb{P}_2$ is given by:

$$W_p(\mathbb{P}_1, \mathbb{P}_2) = \inf_{\gamma \in \mathcal{M}(\mathcal{Y}^2)} \left(\int_{\mathcal{Y} \times \mathcal{Y}} \|y_1 - y_2\|^p \gamma(dy_1, dy_2) \right)^{\frac{1}{p}},$$

where γ is a joint distribution of y_1 and y_2 with marginals $\mathbb{P}_1$ and $\mathbb{P}_2$. They provide conditions to obtain asymptotic convergence and out-of-sample guarantees on the solutions of the DRO model.

Chen and Paschalidis (2019) study a distributionally robust kNN regression problem by combining point estimation of the outcome with a DRO model over a Wasserstein ambiguity set (Chen and Paschalidis 2018) and then using kNN to predict the outcome based on the weighted distance metric constructed from the estimates. Extending the methods developed in Nguyen et al. (2020), Nguyen et al. (2021) study a distributionally robust contextual portfolio allocation problem where worst-case conditional return-risk tradeoff is computed over an optimal transport ambiguity set consisting of perturbations of the joint distribution of covariates and returns. Their approach generalizes the mean-variance and mean-CVaR model, for which the distributionally robust models are shown to be equivalent to semi-definite or second-order cone representable programs. Esteban-Pérez and Morales (2022) solve a DRO problem with a novel ambiguity set that is based on trimming the empirical conditional distribution, i.e., reducing the weights over the support points. The authors show the link between trimming a distribution and partial mass transportation problem, and an interested reader can refer to Esteban-Pérez and Morales (2023) for an application in the optimal power flow problem.

A distributionally robust extension of the `rCSO` model is presented in Kannan et al. (2021) and Kannan et al. (2022). It hedges against all distributions that lie in the r radius of the (Wasserstein) ambiguity ball centered at the estimated distribution $\mathbb{P}^{\text{ER}}(\boldsymbol{x})$. Perakis et al. (2023) propose a DRO model to solve a two-stage multi-item joint production and pricing problem with a partitioned-moment-based ambiguity set constructed by clustering the residuals estimated from an additive demand model.

Zhu et al. (2022) considers an expected value-based model and suggests an ambiguity set that is informed by the estimation metric used to train $g_{\hat{\boldsymbol{\theta}}}$. Namely, they consider:

$$\min_{\boldsymbol{z} \in \mathcal{Z}} \sup_{\boldsymbol{\theta} \in \mathcal{U}(\hat{\boldsymbol{\theta}}, r)} c(\boldsymbol{z}, g_{\boldsymbol{\theta}}(\boldsymbol{x})),$$

with

$$\mathcal{U}(\hat{\boldsymbol{\theta}}, r) := \{\boldsymbol{\theta} \in \boldsymbol{\theta} | \rho(g_{\boldsymbol{\theta}}, \hat{\mathbb{P}}_N) \leq \rho(g_{\hat{\boldsymbol{\theta}}}, \hat{\mathbb{P}}_N) + r\}.$$

They show how finite-dimensional convex reformulations can be obtained when $g_{\boldsymbol{\theta}}(\boldsymbol{x}) := \boldsymbol{\theta}^T \boldsymbol{x}$, and promote the use of a "robustness optimization" form.

5 Integrated learning and optimization

As discussed previously, ILO is an end-to-end framework that includes three components in the training pipeline: (i) a prediction model that maps the covariate to a predicted distribution (or possibly a point prediction), (ii) an optimization model that takes as input a prediction and returns a decision, and

Table 3: Overview of contextual optimization papers in the ILO framework.

	Objective				Feasible domain		Training		
	LP	QP	Convex	Non convex	Integer	Uncertain	Implicit diff.	Surr. loss	Surr. optim.
Donti et al. (2017)	✗	✔	✔	✗	✗	✔	✔	✗	✗
Butler and Kwon (2023a)	✗	✔	✗	✗	✗	✗	✔	✗	✗
McKenzie et al. (2023)	✔	✗	✗	✗	✗	✗	✔	✗	✗
Kotary et al. (2023)	✔	✔	✔	✔	✗	✗	✔	✗	✗
Sun et al. (2023b)	✗	✔	✗	✔	✗	✗	✔	✗	✗
Wilder et al. (2019a)	✔	✗	✗	✗	✗	✗	✔	✗	✗
Ferber et al. (2020)	✔	✗	✗	✗	✔	✗	✔	✗	✗
Mandi and Guns (2020)	✔	✗	✗	✗	✔	✗	✔	✗	✗
Elmachtoub and Grigas (2022)	✔	✗	✗	✗	✔	✗	✗	✔	✗
Loke et al. (2022)	✔	✗	✗	✗	✗	✗	✗	✔	✗
Mandi et al. (2020)	✔	✗	✗	✗	✔	✗	✗	✔	✗
Jeong et al. (2022)	✔	✗	✗	✗	✔	✗	✗	✔	✗
Muñoz et al. (2022)	✔	✗	✗	✗	✗	✗	✗	✔	✗
Estes and Richard (2023)	✔	✗	✔	✗	✗	✗	✗	✔	✗
Elmachtoub et al. (2020)	✔	✗	✗	✗	✗	✗	✗	✔	✗
Kallus and Mao (2022)	✗	✗	✔	✗	✗	✗	✗	✔	✗
Wilder et al. (2019b)	✗	✔	✗	✗	✔	✗	✗	✔	✗
Vlastelica et al. (2019)	✔	✗	✗	✗	✔	✗	✗	✔	✗
Chung et al. (2022)	✗	✗	✔	✗	✔	✗	✗	✔	✗
Lawless and Zhou (2022)	✔	✗	✗	✗	✔	✗	✗	✔	✗
Berthet et al. (2020)	✔	✗	✗	✗	✔	✗	✗	✗	✔
Dalle et al. (2022)	✔	✗	✗	✗	✔	✗	✗	✗	✔
Kong et al. (2022)	✗	✔	✔	✔	✔	✗	✗	✗	✔
Mandi et al. (2022)	✔	✗	✗	✗	✔	✗	✗	✗	✔
Grigas et al. (2021)	✗	✗	✔	✗	✗	✗	✗	✗	✔
Cristian et al. (2022)	✗	✗	✔	✗	✗	✗	✗	✗	✔
Shah et al. (2022)	✗	✔	✔	✔	✔	✗	✗	✗	✔

Notes: an approach has a "Convex" objective if it can handle general convex objective functions that are not linear or quadratic such as convex piecewise-linear objective functions; an "Uncertain" feasible domain denotes that some constraints are subject to uncertainty. Implicit diff., surr. loss and surr. optim. denote implicit differentiation, surrogate differentiable loss function and surrogate differentiable optimizer, respectively.

(iii) a task-based loss function that captures the downstream optimization problem. The parameters of the prediction model are trained to maximize the prescriptive performance of the policy, i.e., it is trained on the task loss incurred by this induced policy rather than the estimation loss.

Next, we discuss several methods for implementing the ILO approach. We start by describing the different models that are used in ILO (Section 5.1), and then we present the algorithms used to perform the training. We divide the algorithms into four categories. Namely, training using unrolling (Section 5.2), implicit differentiation (Section 5.3), a surrogate differentiable loss function (Section 5.4), and a differentiable optimizer (Section 5.5). An overview of the methods presented in this section is given in Table 3.

5.1 Models

Bengio (1997) appears to have been the first to have trained a prediction model using a loss that is influenced by the performance of the decision prescribed by a conditional expected value-based decision rule. This was done in the context of portfolio management, where an investment decision rule exploits a point prediction of asset returns. Effective wealth accumulation is used to steer the predictor toward predictions that lead to good investments. More recent works attempt to integrate a full optimization model, rather than a rule, into the training pipeline. Next, we summarize how ILO is applied to the two types of contextual optimization models and introduce two additional popular task models that have been considered under ILO, replacing the traditional expected cost task.

Expected value-based model. To this date, most of the literature has considered performing ILO on an expected value-based optimization model. Namely, following the notation presented in Definition 1 (Section 2.2.2), this training pipeline is interested in the loss $\mathcal{L}(\boldsymbol{\theta}) := H(z^*(\cdot, g_{\boldsymbol{\theta}}), \hat{\mathbb{P}}_N) = \mathbb{E}_{\hat{\mathbb{P}}_N}[c(z^*(\boldsymbol{x}, g_{\boldsymbol{\theta}}), \boldsymbol{y})]$ with $g_{\boldsymbol{\theta}}(\boldsymbol{x})$ as a point predictor for $\boldsymbol{y}$, which we interpret as a prediction of $\mathbb{E}[\boldsymbol{y}|\boldsymbol{x}]$. This already raises challenges related to the non-convexity of the integrated loss function $\mathcal{L}(\boldsymbol{\theta})$ and its differentiation with respect to $\boldsymbol{\theta}$:

$$\begin{aligned}\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}) &= \frac{1}{N}\sum_{i=1}^{N}\nabla_{\boldsymbol{\theta}} c(z^*(\boldsymbol{x}_i, g_{\boldsymbol{\theta}}), \boldsymbol{y}_i) \\ &= \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{n_z}\sum_{k=1}^{n_y}\frac{\partial c(z^*(\boldsymbol{x}_i, g_{\boldsymbol{\theta}}), \boldsymbol{y}_i)}{\partial z_j}\left.\frac{\partial z_j^*(\boldsymbol{x}_i, \hat{\boldsymbol{y}})}{\partial \hat{y}_k}\right|_{\hat{\boldsymbol{y}}=g_\theta(\boldsymbol{x}_i)}\nabla_{\boldsymbol{\theta}}[g_{\boldsymbol{\theta}}(\boldsymbol{x}_i)]_k\end{aligned}$$

with $\frac{\partial z_j^*(\boldsymbol{x}_i, \hat{\boldsymbol{y}})}{\partial \hat{y}_k}$ as the most problematic evaluation. For instance, when $z^*(\boldsymbol{x}_i, g_{\boldsymbol{\theta}})$ is the solution of a linear program (LP), it is well known that its gradient is either null or non-existent as it jumps between extreme points of the feasible polyhedron as the objective is perturbed.

Conditional distribution-based model. In the context of learning a conditional distribution model $f_\theta(\boldsymbol{x})$, Donti et al. (2017) appear to be the first to study the ILO problem. They model the distribution of the uncertain parameters using parametric distributions (exponential and normal). For the newsvendor problem, it is shown that the ILO model outperforms decision rule optimization with neural networks and SLO with maximum likelihood estimation (MLE) when there is model misspecification. Since then, it has become more common to formulate the CSO problem as a weighted SAA model (as discussed in Section 4.1.2). The prediction model f_θ then amounts to identifying a vector of weights to assign to each historical sample, which effectively reduces to the expected value paradigm with $z^*(\boldsymbol{x}, g_{\boldsymbol{\theta}}) := z^*(\boldsymbol{x}, \sum_{i=1}^N [g_{\boldsymbol{\theta}}(\cdot)]_i \delta_{\boldsymbol{y}_i})$, where $[g_{\boldsymbol{\theta}}(\boldsymbol{x})]_i$ denotes the estimated conditional probability of scenario $\boldsymbol{y}_i$ given $\boldsymbol{x}$. This is done by Kallus and Mao (2022) using a random forest to assign weights and by Grigas et al. (2021) with a neural network to predict the probabilities for a finitely supported $\boldsymbol{y}$.

Regret minimization task. A recent line of work has tackled the ILO problem from the point of view of regret. Indeed, in Elmachtoub and Grigas (2022), a contextual point predictor $g_{\boldsymbol{\theta}}(\boldsymbol{x})$ is learned by minimizing the regret associated with implementing the prescribed decision based on the mean estimator $g_{\boldsymbol{\theta}}(\boldsymbol{x})$ instead of based on the realized parameters $\boldsymbol{y}$ (a.k.a. the optimal hindsight or wait-and-see decision). Specifically, the value of an expected value-based policy $\pi_{\boldsymbol{\theta}}(\boldsymbol{x}) := z^*(\boldsymbol{x}, g_{\boldsymbol{\theta}})$ is measured as the expected regret

$$H_{\text{Regret}}(\pi_{\boldsymbol{\theta}}, \mathbb{P}) := \mathbb{E}_{\mathbb{P}}[c(\pi_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y}) - c(z^*(\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{y})], \tag{12}$$

which returns the same optimal parameter vector $\boldsymbol{\theta}$ as the ILO problem (9). This is due to the fact that:

$$H_{\text{Regret}}(\pi, \hat{\mathbb{P}}_N) = \mathbb{E}_{\hat{\mathbb{P}}_N}[c(\pi(\boldsymbol{x}), \boldsymbol{y}) - c(z^*(\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{y})] = H(\pi, \hat{\mathbb{P}}_N) - \mathbb{E}_{\hat{\mathbb{P}}_N}[c(z^*(\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{y})].$$

Hence, both $H_{\text{Regret}}(\pi, \hat{\mathbb{P}}_N)$ and $H(\pi, \hat{\mathbb{P}}_N)$ have the same set of minimizers.

Optimal action imitation task. ILO has some connections to inverse optimization, i.e., the problem of learning the parameters of an optimization model given data about its optimal solution (see Sun et al. 2023a, where both problems are addressed using the same method). Indeed, one can replace the original objective of ILO with an objective that seeks to produce a $\boldsymbol{z}^*(\boldsymbol{x}, f_{\boldsymbol{\theta}})$ that is as close as possible to the optimal hindsight action and, therefore, closer to the regret objective. Specifically, to learn a policy that "imitates" the optimal hind-sight action, one can first augment the data set with $\boldsymbol{z}_i^* := \boldsymbol{z}^*(\boldsymbol{x}_i, \boldsymbol{y}_i)$ to get $\{(\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{z}_i^*)\}_{i=1}^N$. Thereafter, a prediction model $f_{\boldsymbol{\theta}}(x)$ is learned in a way that the decision $z^*(\boldsymbol{x}_i, f_{\boldsymbol{\theta}})$ is as close as possible to $\boldsymbol{z}_i^*$ for all samples in the training set (Kong et al. 2022):

$$H_{\text{Imitation}}(\pi, \hat{\mathbb{P}}'_N) := \mathbb{E}_{\hat{\mathbb{P}}'_N}[d(\pi(\boldsymbol{x}), \boldsymbol{z}^*)] = \mathbb{E}_{\hat{\mathbb{P}}_N}[d(\pi(\boldsymbol{x}), \boldsymbol{z}^*(\boldsymbol{x}, \boldsymbol{y}))] \tag{13}$$

with $\hat{\mathbb{P}}'_N$ as the empirical distribution on the lifted tuple $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}^*(\boldsymbol{x}, \boldsymbol{y}))$ based on the augmented data set and a distance function $d(\boldsymbol{z}, \boldsymbol{z}^*)$. We note that there is no reason to believe that the best imitator under a general distance function, e.g., $\|\boldsymbol{z} - \boldsymbol{z}^*\|_2$, performs well under our original metric $H(\pi, \hat{\mathbb{P}}_N)$. One exception is for $d(\boldsymbol{z}, \boldsymbol{z}^*) := c(\boldsymbol{z}, \boldsymbol{y}) - c(\boldsymbol{z}^*, \boldsymbol{y})$, where we allow the distance to also depend on $\boldsymbol{y}$, for which we recover the regret minimization approach, and therefore the same solution as with $H(\pi, \hat{\mathbb{P}}_N)$.

5.2 Training by unrolling

An approach to obtain the Jacobian matrix $\frac{\partial \boldsymbol{z}^*(\boldsymbol{x}, \hat{\boldsymbol{y}})}{\partial \hat{\boldsymbol{y}}}$ is unrolling (Domke 2012), which involves approximating the optimization problem with an iterative solver (e.g., first-order gradient-based method). Each operation is stored on the computational graph, which then allows, in principle, for computing gradients through classical back-propagation methods. Unfortunately, this approach requires extensive amounts of memory. Besides this, the large size of the computational graph exacerbates the vanishing and exploding gradient problems typically associated with training neural networks (Monga et al. 2021).

5.3 Training using implicit differentiation

Implicit differentiation allows for a memory-efficient backpropagation as opposed to unrolling (we refer to Bai et al. 2019, for discussion on training constant memory implicit models using an FP equation and feedforward networks of infinite depths). Amos and Kolter (2017) appears to be the first to have employed implicit differentiation methods to train an ILO model, which they refer to as `OptNet`. They consider expected value-based optimization models that take the form of constrained quadratic programs (QP) with equality and inequality constraints. They show how the implicit function theorem (IFT) (see Krantz and Parks 2002) can be used to differentiate $z^*(\boldsymbol{x}, g_{\boldsymbol{\theta}})$ with respect to $\boldsymbol{\theta}$ using the Karush–Kuhn–Tucker (KKT) conditions that are satisfied at optimality. Further, they provide a custom solver based on a primal-dual interior method to simultaneously solve multiple QPs on GPUs in batch form, permitting 100-times speedups compared to Gurobi and CPLEX. This approach is extended to conditional stochastic and strongly convex optimization models in Donti et al. (2017). They use sequential quadratic programming (`SQP`) to obtain quadratic approximations of the objective functions of the convex program at each iteration until convergence to the solution, and then differentiate the last iteration of `SQP` to obtain the Jacobian. For a broader view of implicit differentiation, we refer to the surveys by Duvenaud et al. (2020) and Blondel et al. (2022).

To solve large-scale QPs with linear equality and box inequality constraints, Butler and Kwon (2023a) use the ADMM algorithm to decouple the differentiation procedure for primal and dual variables, thereby decomposing the large problem into smaller subproblems. Their procedure relies on implicit differentiation of the fixed-point (FP) equations of the alternating direction method of multipliers (ADMM) algorithm (ADMM-FP). They show that the unrolling of the iterations of the ADMM algorithm on the computational graph (Sun et al. 2016, Xie et al. 2019) results in higher computation time than ADMM-FP. Their empirical results on a portfolio optimization problem with 254 assets suggest that computational time can be reduced by a factor of almost five by using `ADMM-FP` compared to `OptNet`, mostly due to the use of the ADMM algorithm in the forward pass. Note that the experiments in Butler and Kwon (2023a) were conducted on a CPU.

To extend `OptNet` to a broader class of problems, Agrawal et al. (2019) introduce `CvxpyLayer` that relies on converting disciplined convex programs in the domain-specific language used by Cvxpy into conic programs. They implicitly differentiate the residual map of the homogeneous self-dual embedding associated with the conic program.

McKenzie et al. (2023) note that using KKT conditions for constrained optimization problems with DNN-based policies is computationally costly as "`CvxpyLayer` struggles with solving problems containing more than 100 variables" (see also Butler and Kwon 2023a). An alternative is to use projected gradient descent (`PGD`) where DNN-based policies are updated using an iterative solver and projected onto the constraint set $\mathcal{Z}$ at each iteration and the associated fixed point system (Donti et al. 2021, Chen et al. 2021, Blondel et al. 2022) is used to obtain the Jacobian.

Since a closed-form solution for the projection onto $\mathcal{Z}$ is unavailable in many cases, the projection step may be costly, and in some cases, `PGD` may not even converge to a feasible point (Rychener and Sutter 2023). To avoid computing the projection in the forward pass, McKenzie et al. (2023) solve the expected value-based CSO problem using Davis-Yin operator splitting (Davis and Yin 2017) while the backward pass uses the Jacobian-free backpropagation (JFB – Fung et al. 2022) in which the Jacobian matrix is replaced with an identity matrix.

To mitigate the issues with unrolling, Kotary et al. (2023) propose FP folding (`fold-opt`) that allows analytically differentiating the FP system of general iterative solvers, e.g., `ADMM`, `SQP`, and `PGD`. By unfolding (i.e., partial unrolling), some of the steps of unrolling are grouped in analytically differentiable update function $\mathcal{T} : \mathbb{R}^{n_{\boldsymbol{y}}} \to \mathbb{R}^{n_{\boldsymbol{y}}}$:

$$\boldsymbol{z}_{k+1}(\boldsymbol{x}, \hat{\boldsymbol{y}}) = \mathcal{T}(\boldsymbol{z}_k(\boldsymbol{x}, \hat{\boldsymbol{y}}), \hat{\boldsymbol{y}}).$$

Realizing that $\boldsymbol{z}^*(\boldsymbol{x}, \hat{\boldsymbol{y}})$ is the FP of the above system, they use the IFT to obtain a linear system (a differential FP condition) that can be solved to obtain the Jacobian. This effectively decouples the forward and backward pass enabling the use of black box solvers like Gurobi for the forward pass while `CvxpyLayer` is restricted to operator splitting solvers like ADMM. An added benefit of using `fold-opt` is that it can solve non-convex problems. In the case of portfolio optimization, the authors note that the superior performance of their model with respect to `CvxpyLayer` can be explained by the precise calculations made in the forward pass by Gurobi.

While speedups can be obtained for sparse problems, Sun et al. (2023b) remark that the complexity associated with differentiating the KKT conditions is cubic in the total number of decision variables and constraints in general. They propose an alternating differentiation framework (called `Alt-Diff`) to solve parameterized convex optimization problems with polyhedral constraints using ADMM that decouples the objective and constraints. This procedure results in a smaller Jacobian matrix when there are many constraints since the gradient computations for primal, dual, and slack variables are done alternatingly. The gradients are shown to converge to those obtained by differentiating the KKT conditions. The authors employ truncation of iterations to compensate for the slow convergence of ADMM when compared to interior-point methods and provide theoretical upper bounds on the error in the resulting gradients. `Alt-Diff` is shown to achieve the same accuracy with truncation and lower computational time when compared to `CvxpyLayer` for an energy generation scheduling problem.

Motivated by `OptNet`, several extensions have been proposed to solve linear and combinatorial problems. Wilder et al. (2019a) solve LP-representable combinatorial optimization problems and LP relaxations of combinatorial problems during the training phase. Their model, referred to as `QPTL` (Quadratic Programming Task Loss), adds a quadratic penalty term to the objective function of the linear problem. This has two advantages: it recovers a differentiable linear-quadratic program, and the added term acts as a regularizer, which might avoid overfitting. To solve a general mixed-integer LP (MILP), Ferber et al. (2020) develop a cutting plane method `MIPaal`, which adds a given number of cutting planes in the form of constraints $S\boldsymbol{z} \leq \boldsymbol{s}$ to the LP relaxation of the MILP. Instead of adding a quadratic term, Mandi and Guns (2020) propose `IntOpt` based on the interior point method to solve

LPs that adds a log barrier term to the objective function and differentiates the homogeneous self-dual formulation of the LP. Their experimental analyses show that this approach performs better on energy cost-aware scheduling problems than `QPTL` using the data from Ifrim et al. (2012).

5.4 Training using a surrogate differentiable loss function

As discussed in Section 5.1, minimizing directly the task loss in (9) or the regret in (12) is computationally difficult in most cases. For instance, the loss may be piecewise-constant as a function of the parameters of a prediction model and, thus, may have no informative gradient. To address this issue, several surrogate loss functions with good properties, e.g., differentiability and convexity, have been proposed to train ILO models.

5.4.1 `SPO+`.

In the context of Smart "Predict, then Optimize" (`SPO`), Elmachtoub and Grigas (2022) first tackles the potential non-uniqueness of $z^*(\boldsymbol{x}, g_{\boldsymbol{\theta}})$ by choosing to minimize the empirical average of the regret under the worst-case optimal solution as defined below:

$$\begin{aligned}(\texttt{SPO})\quad & \min_{\boldsymbol{\theta}} \max_{\pi} H_{\text{Regret}}(\pi, \hat{\mathbb{P}}_N), \\ & \text{s.t.}\ \ \pi(\boldsymbol{x}) \in \operatorname*{argmin}_{\boldsymbol{z}\in\mathcal{Z}} c(\boldsymbol{z}, g_{\boldsymbol{\theta}}(\boldsymbol{x})),\ \forall \boldsymbol{x}.\end{aligned} \tag{14}$$

In the expected value-based model, they show that the `SPO` objective reduces to training the prediction model according to the ERM problem:

$$\boldsymbol{\theta}^\star \in \operatorname*{argmin}_{\boldsymbol{\theta}} \rho_{\texttt{SPO}}(g_{\boldsymbol{\theta}}, \hat{\mathbb{P}}_N) := \mathbb{E}_{\hat{\mathbb{P}}_N}\left[\ell_{\texttt{SPO}}(g_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y})\right],$$

with:

$$\ell_{\texttt{SPO}}(\hat{\boldsymbol{y}}, \boldsymbol{y}) := \sup_{\bar{\boldsymbol{z}} \in \operatorname{argmin}_{\boldsymbol{z}\in\mathcal{Z}} c(\boldsymbol{z}, \hat{\boldsymbol{y}})} c(\bar{\boldsymbol{z}}, \boldsymbol{y}) - c(z^*(\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{y}).$$

Since the `SPO` loss function is nonconvex and discontinuous in $\hat{\boldsymbol{y}}$ (Ho-Nguyen and Kılınç-Karzan 2022, Lemma 1), Elmachtoub and Grigas (2022) focus on the linear objective $c(\boldsymbol{z}, \boldsymbol{y}) := \boldsymbol{y}^T\boldsymbol{x}$ and replace the `SPO` loss with a convex envelope approximation called `SPO+` which has a closed-form expression for its subgradient:

$$\ell_{\texttt{SPO+}}(\hat{\boldsymbol{y}}, \boldsymbol{y}) := \sup_{\boldsymbol{z}\in\mathcal{Z}} (\boldsymbol{y} - 2\hat{\boldsymbol{y}})^T\boldsymbol{z} + 2\hat{\boldsymbol{y}}^T\boldsymbol{z}^*(\boldsymbol{x}, \boldsymbol{y}) - \boldsymbol{y}^T\boldsymbol{z}^*(\boldsymbol{x}, \boldsymbol{y}).$$

Loke et al. (2022) propose a decision-driven regularization model (`DDR`) that combines prediction accuracy and decision quality in a single optimization problem with loss function as follows:

$$\ell_{\texttt{DDR}}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = d(\hat{\boldsymbol{y}}, \boldsymbol{y}) - \lambda \min_{\boldsymbol{z}\in\mathcal{Z}}\{\mu\boldsymbol{y}^\top\boldsymbol{z} + (1-\mu)\hat{\boldsymbol{y}}^\top\boldsymbol{z}\}$$

and `SPO+` being a special case with $\mu = -1,\ \lambda = 1$, and $d(\hat{\boldsymbol{y}}, \boldsymbol{y}) = 2\hat{\boldsymbol{y}}^\top z^*(\boldsymbol{x}, \boldsymbol{y}) - \boldsymbol{y}^T\boldsymbol{z}^*(\boldsymbol{x}, \boldsymbol{y})$.

`SPO+` for combinatorial problems. Evaluating the `SPO+` loss requires solving the optimization problem (8) to obtain $\boldsymbol{z}^*(\boldsymbol{x}, 2\hat{\boldsymbol{y}} - \boldsymbol{y})$ for each data point. This can be computationally demanding when the optimization model in (8) is an NP-hard problem. Mandi et al. (2020) propose a `SPO-relax` approach that computes the gradient of `SPO+` loss by solving instead a continuous relaxation when (8) is a MILP. They also suggest speeding up the resolution using a warm-start for learning with a pre-trained model that uses MSE as the loss function. Another way proposed to speed up the computation is warm-starting the solver that is used, e.g., $z^*(\boldsymbol{x}, \boldsymbol{y})$ can be used as a starting point for MILP solvers or to cut away a large part of the feasible space. Mandi et al. (2020) show that for weighted and unweighted knapsack problems as well as energy-cost aware scheduling problems (CSPLib, Problem 059, Simonis et al. 2014), `SPO-relax` results in faster convergence and similar performance compared to `SPO+` loss. Also, `SPO-relax` provides low regret solutions and faster convergence compared to `QPTL` in the aforementioned three problems, except in the weighted knapsack problem with low capacity.

With a focus on exact solution approaches, Jeong et al. (2022) study the problem of minimizing the regret in (12) assuming a linear prediction model $g_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{\theta}\boldsymbol{x}$ with $\boldsymbol{\theta} \in \mathbb{R}^{n_z \times n_x}$. Under the assumption that $\boldsymbol{z}^*(\boldsymbol{x}, g_{\boldsymbol{\theta}})$ is unique for all $\boldsymbol{\theta}$ and $\boldsymbol{x}$, the authors reformulate the bilevel `SPO` problem as a single-level MILP using symbolic variable elimination. They show that their model can achieve up to two orders of magnitude improvement in expected regret compared to `SPO+`. Muñoz et al. (2022) applies a similar idea of representing the set of optimal solutions with a MILP. They rely on the KKT conditions of the problem defining $z^*(\boldsymbol{x}, g_{\boldsymbol{\theta}})$ to transform the bilevel integrated problem into a single-level MILP. Finally, Estes and Richard (2023) use `SPO` loss function to solve a two-stage LP with right-hand side uncertainty. They propose a lexicographical ordering rule to select the minimal solution when there are multiple optima and approximate the resulting piecewise-linear loss function, `lex-SPO`, by a convex surrogate to find the point predictor.

`SPO` Trees. Elmachtoub et al. (2020) propose a model (`SPOT`) to construct decision trees that segment the contextual features based on the `SPO` loss function while retaining the interpretability in the end-to-end learning framework. Their model outperforms classification and regression trees (CART) in the numerical experiments on a news recommendation problem using the Yahoo! Front Page Today Module dataset and on the shortest path problem with synthetic data used in Elmachtoub and Grigas (2022).

Guarantees. Elmachtoub and Grigas (2022) show that under certain conditions, the minimizers of `SPO` loss, `SPO+` loss and MSE loss are almost always equal to $\mathbb{E}_{\mathbb{P}(\boldsymbol{y}|\boldsymbol{x})}[\boldsymbol{y}]$ given that $\mathbb{E}_{\mathbb{P}(\boldsymbol{y}|\boldsymbol{x})}[\boldsymbol{y}] \in \mathcal{H}$. Thus, `SPO+` is Fisher consistent (see Definition A3 in Appendix 1.1) with respect to `SPO` loss. This means that minimizing the surrogate loss also minimizes the true loss function. Ho-Nguyen and Kılınç-Karzan (2022) show that for a multiclass classification problem, `SPO+` is Fisher inconsistent, while MSE loss is consistent. However, complete knowledge of the distribution is a limitation in practice where the decision maker has access to only the samples from the distribution. As a result, Ho-Nguyen and Kılınç-Karzan (2022) and Liu and Grigas (2021) provide calibration bounds that hold for a class of distributions $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$ and ensure that a lower excess risk of predictor for MSE and `SPO+`, respectively, translates to lower excess `SPO` risk (see Definition A4 in Appendix 1.1).

In many ML applications, one seeks to derive finite-sample guarantees which are given in the form of a generalization bound, i.e., an upper bound on the difference between the true risk of a loss function and its empirical risk estimate for a given sample size N. A generalization bound for `SPO` loss function is given in El Balghiti et al. (2022) (extension of El Balghiti et al. (2019)) based on Rademacher complexity (see Definition A5 in Appendix 1.1) of the `SPO` loss composed with the prediction functions $g_{\boldsymbol{\theta}} \in \mathcal{H}$. More specifically, the bound achieved in El Balghiti et al. (2019) is $\mathrm{O}\left(\sqrt{\frac{\log(N)}{N}}\right)$, and tighter bounds with respect to decision and feature dimension are obtained using `SPO` function's structure and if $\mathcal{Z}$ satisfies a "strength" property. Hu et al. (2022) show that for linear CSO problems, the generalization bound for MSE loss and `SPO` loss is $\mathrm{O}(\sqrt{\frac{1}{N}})$ while faster convergence rates for the SLO model compared to ILO model are obtained under certain low-noise assumptions. Elmachtoub et al. (2023) show that for non-linear optimization problems, SLO models stochastically dominate ILO in terms of their asymptotic optimality gaps when the hypothesis class covers the true distribution. When the model is misspecified, they show that ILO outperforms SLO for a newsvendor problem.

5.4.2 Surrogate loss for a stochastic forest.

Kallus and Mao (2022) propose an algorithm called `StochOptForest`, which generalizes the random-forest based local parameter estimation procedure in Athey et al. (2019). A second-order perturbation analysis of stochastic optimization problems allows them to scale to larger CSO problems since they can avoid solving an optimization problem at each candidate split. The policies obtained using their model are shown to be asymptotically consistent, and the benefit of end-to-end learning is illustrated by comparing their approach to the random forests of Bertsimas and Kallus (2020) on a set of problems with synthetic and real-world data.

5.4.3 Other surrogates.

Wilder et al. (2019b) introduce `ClusterNet` to solve hard combinatorial graph optimization problems by learning incomplete graphs. The model combines graph convolution networks to embed the graphs in a continuous space and uses a soft version of k-means clustering to obtain a differential proxy for the combinatorial problems, e.g., community detection and facility location. Numerical experiments on a synthetic data set show that `ClusterNet` outperforms the two-stage SLO approach of first learning the graph and then optimizing, as well as other baselines used in community detection and facility location.

Focusing on combinatorial problems, Vlastelica et al. (2019) tackles the issue that the Jacobian of $z^*(\boldsymbol{x}, g_{\boldsymbol{\theta}})$ is zero almost everywhere by approximating the true loss function using an interpolation controlled in a way that balances between "informativeness of the gradient" and "faithfulness to the original function". Algorithmically, this is done by perturbing the prediction $g_{\boldsymbol{\theta}}(\boldsymbol{x})$ in the direction $\nabla_{\boldsymbol{z}} c(z^*(\boldsymbol{x}, g_{\boldsymbol{\theta}}), \boldsymbol{y})$ and obtaining a gradient of the surrogate loss based on the effect of this perturbation on the resulting perturbed action.

In Chung et al. (2022), a decision-focused learning-based approach is used to allocate medicines in a health supply chain in Sierra Leone. Using the Taylor expansion of the decision loss and a single prediction model for all the facilities, the resulting decision-aware model is trained using random forests. To improve the scalability compared to Kallus and Mao (2022) and Wang et al. (2023), they propose a weighted average distance over M facilities:

$$d(g_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y}) = \sum_{j=1}^{M} |w_j| . |g_{\boldsymbol{\theta}}(\boldsymbol{x}_j) - \boldsymbol{y}_j|, \tag{15}$$

where the weights are given by $w_j = \left(\nabla_{\boldsymbol{y}} \boldsymbol{z}^*(\boldsymbol{x}, \boldsymbol{y}) \nabla_{\boldsymbol{z}} c(\boldsymbol{z}^*(\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{y})\right)_j$. They train a random forest to minimize this surrogate loss. To recover a weighted SAA formulation, they define the conditional distribution $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{T} \sum_{t=1}^{T} \delta_{g^t_{\boldsymbol{\theta}}(\boldsymbol{x})}$, where $g^t_{\boldsymbol{\theta}}$ is the t-th regression tree of the random forest of size T.

The decision-aware loss in (15) is in line with the expected value-based model in Lawless and Zhou (2022) for linear cost function, i.e., $c = \boldsymbol{y}^\top \boldsymbol{z}$. The prediction error is weighted with a decision-aware regret term as follows:

$$d(g_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y}) = [c(z^*(\boldsymbol{x}, g_{\boldsymbol{\theta}}(\boldsymbol{x})), \boldsymbol{y}) - c(z^*(\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{y})](\boldsymbol{y} - g_{\boldsymbol{\theta}}(\boldsymbol{x}))^2 \tag{16}$$

Learning optimal $\boldsymbol{\theta}$ from the above formulation involves an argmin differentiation. So, the authors provide a two-step polynomial time algorithm to approximately solve the above problem. It first computes a pilot estimator $g_{\hat{\boldsymbol{\theta}}}$ by solving (7) with $d(g_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y}) = (g_{\boldsymbol{\theta}}(\boldsymbol{x}) - \boldsymbol{y})^2$ and then solving (7) with the distance function in (16) where $c(z^*(\boldsymbol{x}, g_{\boldsymbol{\theta}}(\boldsymbol{x})), \boldsymbol{y})$ is substituted with $c(z^*(\boldsymbol{x}, g_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})), \boldsymbol{y})$. The authors show that their simple algorithm performs comparably to `SPO+`.

We conclude this subsection on surrogate loss functions by mentioning the efforts in Sun et al. (2023a) to learn a decision-aware cost point estimator (in an expected value-based model) to imitate the hindsight optimal solution. This is done by designing a surrogate loss function that penalizes by how much the optimal basis optimality conditions are violated. They derive generalization error bounds for this new loss function and employ them to provide a bound on the sub-optimality of the minimal $\boldsymbol{\theta}$.

5.5 Training using a surrogate differentiable optimizer

5.5.1 Differentiable perturbed optimizer.

One way of obtaining a differentiable optimizer is to apply a stochastic perturbation to the parameters predicted by the ML model. Taking the case of expected value-based models as example, the key idea is that, although the gradient of the solution of the conditional problem with respect to the predicted parameters $\hat{\boldsymbol{y}} := g_{\boldsymbol{\theta}}(\boldsymbol{x})$ is zero almost everywhere, if we perturb the predictor using a noise with differentiable density, then the expectation of the solution of the perturbed contextual problem,

$$\bar{z}^{\varepsilon}(\boldsymbol{x}, g_{\boldsymbol{\theta}}) = \mathbb{E}_{\Psi}[\tilde{\boldsymbol{z}}^{\varepsilon}(\boldsymbol{x}, g_{\boldsymbol{\theta}}, \Psi)] \text{ with } \tilde{\boldsymbol{z}}^{\varepsilon}(\boldsymbol{x}, g_{\boldsymbol{\theta}}, \Psi) := \underset{\boldsymbol{z} \in \mathcal{Z}}{\operatorname{argmin}}\, c(\boldsymbol{z}, g_{\boldsymbol{\theta}}(\boldsymbol{x}) + \varepsilon\Psi),$$

where $\varepsilon > 0$ controls the amount of perturbation, and more generally of the expected cost of the associated random policy $\mathbb{E}_{\Psi}[H(\tilde{z}^{\varepsilon}(\cdot, g_{\boldsymbol{\theta}}, \Psi), \hat{\mathbb{P}}_N)]$ can be shown to be smooth and differentiable. This idea is proposed and exploited in Berthet et al. (2020), which focus on a bi-linear cost $c(\boldsymbol{z}, \boldsymbol{y}) := \boldsymbol{y}^T \boldsymbol{z}$ thus simplifying $\mathbb{E}_{\Psi}[H(\tilde{z}^{\varepsilon}(\cdot, g_{\boldsymbol{\theta}}, \Psi), \hat{\mathbb{P}}_N)] = H(\bar{z}^{\varepsilon}(\cdot, g_{\boldsymbol{\theta}}), \hat{\mathbb{P}}_N)$. Further, they show that when an imitation ILO model is used with a special form of Bregman divergence to capture the difference between $\boldsymbol{z}^*(\boldsymbol{x}, \boldsymbol{y})$ and $\tilde{z}^{\varepsilon}(\boldsymbol{x}, \hat{\boldsymbol{y}}, \Psi)$, the gradient of $H_{\text{Imitation}}(\tilde{z}^{\varepsilon}(\cdot, g_{\boldsymbol{\theta}}, \Psi), \hat{\mathbb{P}}'_N)$ can be computed directly without needing to determine the Jacobian of $\bar{\boldsymbol{z}}^{\varepsilon}(\boldsymbol{x}, g_{\boldsymbol{\theta}})$ (Blondel et al. 2020):

$$H_{\text{Imitation}}(\tilde{z}^{\varepsilon}(\cdot, g_{\boldsymbol{\theta}}, \Psi), \hat{\mathbb{P}}'_N) := \mathbb{E}_{\hat{\mathbb{P}}_N}[\ell_{\text{FY}}(g_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y})]$$

with

$$\ell_{\text{FY}}(\hat{\boldsymbol{y}}, \boldsymbol{y}) := \hat{\boldsymbol{y}}^T \boldsymbol{z}^*(\boldsymbol{x}, \boldsymbol{y}) - \mathbb{E}_{\Psi}[(\hat{\boldsymbol{y}} + \varepsilon\Psi)^T \tilde{\boldsymbol{z}}^{\varepsilon}(\boldsymbol{x}, \hat{\boldsymbol{y}}, \Psi)] + \varepsilon\Omega_{\text{FY}}(\boldsymbol{z}^*(\boldsymbol{x}, \boldsymbol{y})),$$

where $\Omega_{\text{FY}}(\boldsymbol{z})$ is the Fenchel dual of $F(\boldsymbol{y}) := -\mathbb{E}_{\Psi}[(\boldsymbol{y} + \Psi)^T \tilde{\boldsymbol{z}}^{\varepsilon}(\boldsymbol{x}, \boldsymbol{y}, \Psi)]$. The gradient of the Fenchel-Young loss with respect to the model prediction is given by:

$$\nabla_{\hat{\boldsymbol{y}}} \ell_{\text{FY}}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \boldsymbol{z}^*(\boldsymbol{x}, \boldsymbol{y}) - \bar{\boldsymbol{z}}^{\varepsilon}(\boldsymbol{x}, \hat{\boldsymbol{y}}).$$

Dalle et al. (2022) introduce a multiplicative perturbation with the advantage that it preserves the sign of $g_{\boldsymbol{\theta}}(\boldsymbol{x})$ without adding any bias:

$$\tilde{\boldsymbol{z}}^{\varepsilon}(\boldsymbol{x}, g_{\boldsymbol{\theta}}, \Psi) := \underset{\boldsymbol{z} \in \mathcal{Z}}{\operatorname{argmin}}\, c(\boldsymbol{z}, g_{\boldsymbol{\theta}}(\boldsymbol{x}) \odot \exp(\varepsilon\Psi - \varepsilon^2/2)),$$

where $\odot$ is the Hadamard dot-product and the exponential is taken elementwise. Dalle et al. (2022) and Sun et al. (2023c) also show that there is a one-to-one equivalence between the perturbed optimizer approach and using a regularized randomized version of the CSO problem for combinatorial problems with linear objective functions. Finally, Dalle et al. (2022) show an intimate connection between the perturbed minimizer approach proposed by Berthet et al. (2020) and surrogate loss functions approaches such as `SPO+` by casting them as special cases of a more general surrogate loss formulation.

Mulamba et al. (2021) and Kong et al. (2022) consider an "energy-based" perturbed optimizer defined by its density of the form

$$\tilde{\boldsymbol{z}}^{\varepsilon}(\boldsymbol{x}, f_{\boldsymbol{\theta}}) \sim \frac{\exp(-h(\boldsymbol{z}, f_{\boldsymbol{\theta}}(\boldsymbol{x}))/\varepsilon)}{\int \exp(-h(\boldsymbol{z}', f_{\boldsymbol{\theta}}(\boldsymbol{x}))/\varepsilon) d\boldsymbol{z}'},$$

with $\varepsilon = 1$, in the context of an imitation ILO problem. This general form of perturbed optimizer captures a varying amount of perturbation through ε, with $\tilde{\boldsymbol{z}}^{\varepsilon}(\boldsymbol{x}, f_{\boldsymbol{\theta}})$ converging in distribution to $\boldsymbol{z}^*(\boldsymbol{x}, f_{\boldsymbol{\theta}})$ as ε goes to zero. They employ the negative log-likelihood to measure the divergence between $\tilde{\boldsymbol{z}}^{\varepsilon}(\boldsymbol{x}, f_{\boldsymbol{\theta}})$ and the hindsight optimal solution $\boldsymbol{z}^*(\boldsymbol{x}, \boldsymbol{y})$. Given the difficulties associated with calculating the partition function in the denominator of (5.5.1), Mulamba et al. (2021) devise a surrogate loss function based on noise-contrastive estimation, which replaces likelihood with relative likelihood when compared to a set of sampled suboptimal solutions. This scheme is shown to improve the performance over `SPO+` and `DBB` in terms of expected regret performance for linear combinatorial CSO. Based on the noise contrastive estimation approach of Mulamba et al. (2021), Mandi et al. (2022) note that ILO for combinatorial problems can be viewed as a learning-to-rank problem. They propose surrogate loss functions, with closed-form expressions for gradients, that are used to train to rank feasible points in terms of performance on the downstream optimization problem. Unlike Mulamba et al. (2021), Kong et al. (2022) tackles the partition function challenge by employing a self-normalized importance sampler that provides a discrete approximation. To avoid overfitting, the authors also introduce a regularization that penalizes the KL divergence between the perturbed optimizer distribution and a subjective posterior distribution over perturbed optimal hindsight actions $\mathbb{P}(\tilde{\boldsymbol{z}}^{\varepsilon}(\boldsymbol{x},\boldsymbol{y})|\boldsymbol{y})$:

$$
\begin{aligned}
&H_{\text{Imitation}}(\tilde{\boldsymbol{z}}^{\varepsilon}(\cdot, f_{\boldsymbol{\theta}}), \hat{\mathbb{P}}'_N) := \\
&\qquad - \mathbb{E}_{\hat{\mathbb{P}}_N}[\log(\mathbb{P}(\tilde{\boldsymbol{z}}^{\varepsilon}(\boldsymbol{x}, f_{\boldsymbol{\theta}}) = \boldsymbol{z}^*(\boldsymbol{x},\boldsymbol{y}))|\boldsymbol{x},\boldsymbol{y})] + \lambda \mathbb{E}_{\hat{\mathbb{P}}_N}[\mathrm{KL}(\mathbb{P}(\tilde{\boldsymbol{z}}^{\varepsilon}(\boldsymbol{x},\boldsymbol{y})|\boldsymbol{y}) \| \tilde{\boldsymbol{z}}^{\varepsilon}(\boldsymbol{x}, f_{\boldsymbol{\theta}})|\boldsymbol{x},\boldsymbol{y})].
\end{aligned}
$$

The authors show that their model outperforms ILO trained using `SQP` and `CvxpyLayer` in terms of computational time and gives lower task loss than sequential models trained using MLE and policy learning with neural networks.

5.5.2 Supervised learning.

Grigas et al. (2021) solve a decision regularized CSO problem with a convex and non-negative decision regularizer $\Omega(\boldsymbol{z})$ assuming that the $\boldsymbol{y}$ has discrete support. Their model, called `ICEO-`λ, is thus trained by solving:

$$
\begin{aligned}
(\text{ICEO-}\lambda) \qquad & \min_{\boldsymbol{\theta}} \quad H(\boldsymbol{z}^*_{\lambda}(\cdot, f_{\boldsymbol{\theta}}), \hat{\mathbb{P}}_N) + \lambda \Omega(\boldsymbol{z}) && \text{(17a)}\\
& \ \text{s.t.} \quad z^*_{\lambda}(\boldsymbol{x}, f_{\boldsymbol{\theta}}) = \operatorname*{argmin}_{\boldsymbol{z}} c(z, f_{\boldsymbol{\theta}}(\boldsymbol{x})) + \lambda \Omega(\boldsymbol{z}). && \text{(17b)}
\end{aligned}
$$

The regularization ensures uniqueness and Lipschitz property of $z^*_{\lambda}(\boldsymbol{x}, f_{\boldsymbol{\theta}})$ with respect to $f_{\boldsymbol{\theta}}$ and leads to finite-sample guarantees. To circumvent the challenge associated with non-differentiability of $z^*_{\lambda}(\boldsymbol{x}, f_{\boldsymbol{\theta}})$ with respect to $\boldsymbol{\theta}$, they replace $z^*_{\lambda}(\boldsymbol{x}, f_{\boldsymbol{\theta}})$ with a smooth approximation $\tilde{z}_{\lambda}(\boldsymbol{x}, f_{\boldsymbol{\theta}})$ that is learned using a random data set $(\boldsymbol{p}_i, \boldsymbol{z}_i)$ generated by sampling $\boldsymbol{p}_i$ from the probability simplex over the discrete support and then finding the optimal solution $\boldsymbol{z}_i$. They show asymptotic optimality and consistency of their solutions when the hypothesis class is well-specified. They compare their approach to other ILO pipelines and to the SLO approach that estimates the conditional distribution using cross-entropy.

Cristian et al. (2022) introduce the `ProjectNet` model to solve uncertain constrained linear programs in an end-to-end framework by training an optimal policy network, which employs a differentiable approximation of the step of projection to feasibility.

Another approach, related to Berthet et al. (2020), that generalizes beyond LPs is given in Shah et al. (2022) that constructs locally optimized decision losses (`LODL`) with supervised learning to directly evaluate the performance of the predictors on the downstream optimization task. To learn a convex `LODL` for each data point, this approach first generates labels in the neighborhood of label $\boldsymbol{y}_i$ in the training set, e.g., by adding Gaussian noise, and then chooses the parameter that minimizes the MSE between `LODL` and the downstream decision loss. The `LODL` is used in place of the task-specific surrogate optimization layers and outperforms SLO on three resource allocation problems (linear top-1 item selection problem, web advertising, and portfolio optimization). The numerical experiments indicate that handcrafted surrogate functions only perform better for the web advertising problem.

5.6 Applications

In this subsection, we discuss the applications of the ILO framework to a wide range of real-world problems.

Tian et al. (2023) use SPOT to solve a maritime transportation problem. Stratigakos et al. (2022) propose an integrated forecasting and optimization model for trading in renewable energy that trains an ensemble of prescriptive trees by randomly splitting the feature space $\mathcal{X}$ based on the task-specific cost function. Finally, SPO has been used in solving last-mile delivery (Chu et al. 2023) and ship inspection problems (Yan et al. 2020). Demirović et al. (2019) and Demirović et al. (2020) minimize the same expected regret as SPO for specific applications related to ranking optimization and dynamic programming problems, respectively.

Perrault et al. (2020) solve a Stackelberg security game with the ILO framework by learning the attack probability distribution over a discrete set of targets to maximize a surrogate for the defender's expected utility. They show that their model results in higher expected utility for the defender on synthetic and human subjects data than the sequential models that learn the attack probability by minimizing the cross entropy loss. Wang et al. (2020) replace the large-scale optimization problem with a low dimensional surrogate by reparameterizing the feasible space of decisions. They observe significant performance improvements for non-convex problems compared to the strongly convex case.

Sang et al. (2022) introduce a decision-focused electricity price prediction approach for energy storage system arbitrage. They present a hybrid loss function to measure prediction and decision errors and a hybrid stochastic gradient descent learning method. Sang et al. (2023) solve a voltage regulation problem using a similar hybrid loss function, and backpropagation is done by implicitly differentiating the optimality conditions of a second-order cone program.

Liu et al. (2023b) use DNN to model the routing behavior of users in a transportation network and learn the parameters by minimizing the mismatch between the flow prescribed by the variational inequality and the observed flow. The backward pass is obtained by applying the IFT to the variational inequality. Wahdany et al. (2023) propose an integrated model for wind-power forecasting that learns the parameters of a neural network to optimize the energy system costs under the system constraints. Vohra et al. (2023) apply similar ideas to develop end-to-end renewable energy generation forecasts, using multiple contextual sources such as satellite images and meteorological time series.

Butler and Kwon (2023b) solves the contextual mean-variance portfolio (MVP) optimization problem by learning the parameters of the linear prediction model using the ILO framework. The covariance matrix is estimated using the exponentially weighted moving average model. They provide analytical solutions to unconstrained and equality-constrained MVP optimization problems and show that they outperform SLO models based on OLS. These analytical solutions lead to lower variance when compared with the exact solutions of the corresponding inequality-constrained MVP optimization problem.

6 Conclusion and future research directions

We now summarize the key research directions for further work in contextual optimization.

Uncertainty in constraints. Most studies on contextual optimization assume that there is no uncertainty in the constraints. If constraints are also uncertain, the SAA solutions that ignore the covariates information might not be feasible (Rahimian and Pagnoncelli 2022). Bertsimas and Kallus (2020) have highlighted the challenges in using ERM in a constrained CSO problem. Rahimian and Pagnoncelli (2022) solve a conditional chance-constrained program that ensures with a high probability that the solution remains feasible under the conditional distribution of the features. Although they do not focus on contextual optimization, interesting links can be found with the literature on constraint learning (Fajemisin et al. 2023) and inverse optimization (Chan et al. 2021).

Risk aversion. There has been a growing interest in studying contextual optimization in the risk-averse setting. Specifically, one can consider replacing the risk-neutral expectation from (1) with a risk measure such as value-at-risk. By doing so, one would expect, with a high probability, that a decision maker's loss is lower than a particular threshold. One can easily represent such a risk measure using an uncertainty set which represents the set of all possible outcomes that may occur in the future. The resulting uncertainty set should be carefully chosen. It should capture the most relevant scenarios to balance the trade-off between avoiding risks and obtaining returns. The recently proposed Conditional Robust Optimization (CRO) paradigm by Chenreddy et al. (2022) (see also Ohmori 2021, Sun et al. 2023b, Peršak and Anjos 2023) consists in learning a conditional set $\mathcal{U}(\boldsymbol{x})$ to solve the following problem:

$$\text{(CRO)} \quad \min_{\boldsymbol{z}\in\mathcal{Z}} \max_{\boldsymbol{y}\in\mathcal{U}(\boldsymbol{x})} c(\boldsymbol{z},\boldsymbol{y}), \tag{18}$$

where $\mathcal{U}(\boldsymbol{x})$ is an uncertainty set designed to contain with high probability the realization of y conditionally on observing $\boldsymbol{x}$. Their approach solves the CRO problem sequentially where $\mathcal{U}(\boldsymbol{x})$ is learned first and is subsequently used to solve the downstream RO problem. A challenging problem is to learn the uncertainty set to minimize the downstream cost function.

Toolboxes and benchmarking. Several toolboxes and packages have been proposed recently to train decision pipelines. Agrawal et al. (2019) provide the `CvxpyLayer` library, which includes a subclass of convex optimization problems as differentiable layers in auto-differentiation libraries in PyTorch, TensorFlow, and JAX. Other libraries for differentiating non-linear optimization problems for end-to-end learning include higher (Grefenstette et al. 2019), JAXopt (Blondel et al. 2022), TorchOpt (Ren et al. 2022), and Theseus (Pineda et al. 2022). Tang and Khalil (2022) introduce PyEPO as an open-source software package in Python for ILO of problems that are linear in uncertain parameters. They implement various existing methods, such as `SPO+`, `DBB`, `DPO`, and `FYL`. They also include new benchmarks and comprehensive experiments highlighting the advantages of integrated learning. Dalle et al. (2022) provide similar tools for combinatorial problems in Julia.

Comparisons of existing approaches in fixed simulation settings are scarce, especially with real-world data. Buttler et al. (2022) provide a meta-analysis of selected methods on an unconstrained newsvendor problem on four data sets from the retail and food sectors. They highlight that there is no single method that clearly outperforms all the others on the four data sets.

Endogenous uncertainty. While there has been some progress in studying problems where the decision affects the uncertain parameters (Basciftci et al. 2021), the literature on decision-dependent uncertainty with covariates is sparse (Bertsimas and Kallus 2020, Bertsimas and Koduri 2022). An example could be a facility location problem where demand changes once a facility is located in a region or a price-setting newsvendor problem whose demand depends on the price (Liu and Zhang 2023). In these problems, the causal relationship between demand and prices is unknown. Interesting connections can be drawn with the literature on heterogeneous treatment effects, such as Wager and Athey (2018), who introduce causal forests for estimating treatment effects and provide asymptotic consistency results. Alley et al. (2023) study a price-setting problem and provide a new loss function to isolate the causal effects of price on demand from the conditional effects due to other features.

Data privacy. Another issue is that the data might come from multiple sources and contain sensitive private information, so it cannot be directly provided in its original form to the system operator. Differential privacy techniques (see, e.g., Abadi et al. 2016) can be used to obfuscate data, but may impact predictive and prescriptive performance. Mieth et al. (2023) determine the data quality after obfuscation in an optimal power flow problem with a Wasserstein ambiguity set and use a DRO model to determine the data value for decision-making.

Interpretability & explainability. Decision pipelines must be trusted to be implemented. This is evident from the European Union legislation "General Data Protection Regulation" that requires entities using automated systems to provide "meaningful information about the logic involved" in making decisions, known popularly as the "right to explanation" (Doshi-Velez and Kim 2017, Kaminski 2019). For instance, a citizen has the right to ask a bank for an explanation in the case of loan denial.

While interpretability has received much attention in predictive ML applications (Rudin 2019), it remains largely unexplored in a contextual optimization, i.e., prescriptive context. Interpretability requires transparent decision pipelines that are intelligible to users, e.g., built over simple models such as decision trees or rule lists. In contrast, explainability may be achieved with an additional algorithm on top of a black box or complex model. Feature importance has been analyzed in a prescriptive context by Serrano et al. (2022). They introduce an integrated approach that solves a bilevel program with an integer master problem optimizing (cross-)validation accuracy. To achieve explainability, Forel et al. (2023) adapt the concept of counterfactual explanations to explain a given data-driven decision through differences of context that make this decision optimal, or better suited than a given expert decision. Having identified these differences, it becomes possible to correct or complete the contextual information, if necessary, or otherwise to give explanative elements supporting different decisions.

Fairness. Applying decisions based on contextual information can raise fairness issues when the context is made of protected attributes. This has been studied especially in pricing problems, to prevent that different customers or groups of customers are proposed prices that differ too greatly (Cohen et al. 2021, 2022).

Finite sample guarantees for ILO. An open problem is to derive generalization bounds on the performance of ILO models for non-linear problems.

Multi-agent decision-making. A multi-agent perspective becomes necessary in transportation and operations management problems, where different agents have access to different sources of information (i.e. covariates). In this regard, some recent work by Heaton et al. (2022) identifies the Nash equilibrium of contextual games using implicit differentiation of variational inequalities and JFB.

Costly label acquisition. In many applications, it is costly to obtain the labels for uncertain parameters and covariate vectors. For instance, in personalized pricing, surveys can be sent to customers to obtain information on the sensitivity of purchasing an item with respect to its price. However, creating, sending, and collecting the surveys may have a cost. Liu et al. (2023a) develop an active learning approach to obtain labels to solve the `SPO` problem, while the more general case of developing active learning methods for non-linear contextual optimization is an interesting future direction. Besbes et al. (2023) provide theoretical results on the trade-off between the quality and quantity of data in a newsvendor problem, thus guiding decision-makers on how to invest in data acquisition strategies.

Multi-stage contextual optimization. Most works on contextual optimization focus on single and two-stage problems. Ban et al. (2019) and Rios et al. (2015) use the residuals of the regression model to build multi-stage scenario trees and solve multi-stage CSO problems. Bertsimas et al. (2023) generalize the weighted SAA model for multi-stage problems. Qi et al. (2023) propose an end-to-end learning framework to solve a real-world multistage inventory replenishment problem.

An active area of research is sequential decision-making with uncertainty, where estimates on the transition dynamics and reward functions of the Markov decision processes (MDPs) are obtained through MLE (Rust 1988). Nikishin et al. (2022) introduce a model-based reinforcement learning approach that combines learning and planning to optimize expected returns for both tabular and non-tabular MDPs. They employ the soft version of the Bellman operator (Ziebart et al. 2008) for efficient parameter learning using the IFT and show that their state-action value function has a lower approximation error than that of MLE in tabular MDPs.

Another interesting research direction is to challenge the assumption that the joint distribution of the context and uncertain parameters is stationary. Neghab et al. (2022) study a newsvendor model with a hidden Markov model underlying the distribution of the features and demand.

Finally, an area that requires attention is the deployment of models for real-world applications by tackling computational hurdles associated with decision-focused learning in MDPs, such as large state-action pairs and high-dimensional policy spaces (Wang et al. 2023). An example is a service call scheduling problem that is formulated as a restless multi-armed bandit (RMAB) problem in Mate et al. (2022) to improve maternal and child health in a non-profit organization. They model each beneficiary as an arm, apply a clustering method to learn the dynamics, and then use the Whittle Index policy to solve the RMAB. Wang et al. (2023) use decision-focused learning to solve RMAB, where the computational difficulty in differentiating the Whittle index policy of selecting the top-k arms, is mitigated by making a soft-top-k selection of arms which is an optimal transport problem (Xie et al. 2020).

1 Appendix

1.1 Theoretical guarantees

In this section, we provide some definitions and theoretical results for completeness.

Definition A1. (Bertsimas and Kallus 2020) We say that a policy $\pi^N(\boldsymbol{x})$ obtained using N samples is asymptotically optimal if, with probability 1, we have that for $\mathbb{P}(\boldsymbol{x})$-almost-everywhere $\boldsymbol{x} \in \mathcal{X}$:

$$\lim_{N\to\infty} h(\pi^N(\boldsymbol{x}), \mathbb{P}(\boldsymbol{y}|\boldsymbol{x})) = h(\pi^*(\boldsymbol{x}), \mathbb{P}(\boldsymbol{y}|\boldsymbol{x})).$$

Definition A2. (Bertsimas and Kallus 2020) We say that a policy $\pi^N(\boldsymbol{x})$ obtained using N samples is consistent if, with probability 1, we have that for $\mathbb{P}(\boldsymbol{x})$-almost-everywhere $\boldsymbol{x} \in \mathcal{X}$:

$$\|\pi^N(\boldsymbol{x}) - \mathcal{Z}^*(\boldsymbol{x})\| = 0 \quad \text{where } \|\pi^N(\boldsymbol{x}) - \mathcal{Z}^*(\boldsymbol{x})\| = \inf_{\boldsymbol{z}\in\mathcal{Z}^*(\boldsymbol{x})} \|\pi^N(\boldsymbol{x}) - \boldsymbol{z}\|,$$

and

$$\mathcal{Z}^*(\boldsymbol{x}) = \{\boldsymbol{z}|\boldsymbol{z} \in \operatorname*{argmin}_{\boldsymbol{z}'\in\mathcal{Z}} \mathbb{E}_{\mathbb{P}}\big[h(\boldsymbol{z}', \mathbb{P}(\boldsymbol{y}|\boldsymbol{x}))\big].$$

The above conditions imply that, as the number of samples tends to infinity, the performance of the decision under almost all covariates matches the optimal conditional cost.

Definition A3. A surrogate loss function ℓ is Fisher consistent with respect to the SPO loss if the following condition holds for all $\boldsymbol{x} \in \mathcal{X}$:

$$\operatorname*{argmin}_{\boldsymbol{\theta}} \mathbb{E}_{\mathbb{P}(\boldsymbol{y}|\boldsymbol{x})}[\ell(g_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y})] \subseteq \operatorname*{argmin}_{\boldsymbol{\theta}} \mathbb{E}_{\mathbb{P}(\boldsymbol{y}|\boldsymbol{x})}[\ell_{\texttt{SPO}}(g_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y})].$$

The Fisher consistency condition defined above requires complete knowledge of the joint distribution $\mathbb{P}$. Instead, an interesting issue is to determine if the surrogate loss ℓ is calibrated with respect to `SPO`, that is, whether low surrogate excess risk translates to small excess true risk.

Definition A4. (Ho-Nguyen and Kılınç-Karzan 2022) A loss function ℓ is uniformly calibrated with respect to `SPO` for a class of distributions $\mathcal{D}$ on $\mathcal{X}\times\mathcal{Y}$ if for $\epsilon > 0$, there exists a function $\Delta_\ell(\cdot) : \mathbb{R}^+ \to \mathbb{R}^+$ such that for all $x \in \mathcal{X}$:

$$\begin{aligned}&\mathbb{E}_{\mathbb{P}(\boldsymbol{y}|\boldsymbol{x})}[\ell(g_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y})] - \inf_{\boldsymbol{\theta}'} \mathbb{E}_{\mathbb{P}(\boldsymbol{y}|\boldsymbol{x})}[\ell(g_{\boldsymbol{\theta}'}(\boldsymbol{x}), \boldsymbol{y})] < \Delta_\ell(\epsilon)\\ &\quad\Rightarrow \mathbb{E}_{\mathbb{P}(\boldsymbol{y}|\boldsymbol{x})}[\ell_{\texttt{SPO}}(g_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y})] - \inf_{\boldsymbol{\theta}'} \mathbb{E}_{\mathbb{P}(\boldsymbol{y}|\boldsymbol{x})}[\ell_{\texttt{SPO}}(g_{\boldsymbol{\theta}'}(\boldsymbol{x}), \boldsymbol{y})] < \epsilon.\end{aligned}$$

Ho-Nguyen and Kılınç-Karzan (2022) introduce a "calibration function" with which it is simpler to verify the uniform calibration of MSE loss and show that the calibration function is $\mathrm{O}(\epsilon^2)$. Liu and Grigas (2021) assumed that the conditional distribution of $\boldsymbol{y}$ given $\boldsymbol{x}$ is bounded from below by the density of a normal distribution and obtain a $\mathrm{O}(\epsilon^2)$ calibration function for polyhedral $\mathcal{Z}$ and $\mathrm{O}(\epsilon)$ when $\mathcal{Z}$ is a level set of a strongly convex and smooth function.

Definition A5. The empirical Rademacher complexity of a hypothesis class $\mathcal{H}$ under the loss function ℓ is given by:

$$\mathbb{E}_\sigma \left[\sup_{g \in \mathcal{H}} \frac{1}{N} \left| \sum_{i=1}^{N} \sigma_i \ell(g(\boldsymbol{x}_i), \boldsymbol{y}_i) \right| \right],$$

where $\sigma_1, \sigma_2, \cdots, \sigma_N$ are independent and identically distributed Rademacher random variables, i.e., $\mathbb{P}(\sigma_i = 1) = \mathbb{P}(\sigma_i = -1) = \frac{1}{2}$, $\forall i \in \{1, 2, \cdots, N\}$.

1.2 List of abbreviations

In this section, we provide expand the abbreviations used in this survey.

Table A1: List of abbreviations

Abbreviation	Description
ADMM	alternating direction method of multipliers
CSO	conditional stochastic optimization
CVaR	conditional value at risk
DRO	distributionally robust optimization
DNN	deep neural network
ERM	empirical risk minimization
FP	fixed point
ILO	integrated learning and optimization
IFT	implicit function theorem
JFB	Jacobian-free backpropagation
kNN	k-nearest neighbor
KKT	Karush–Kuhn–Tucker
KL	Kullback-Leibler
LDR	linear decision rule
LP	linear program
ML	machine learning
MLE	maximum likelihood estimation
MILP	mixed-integer linear program
NW	Nadaraya-Watson
rCSO	residual-based conditional stochastic optimization
RKHS	reproducing kernel Hilbert space
SAA	sample average approximation
SLO	sequential learning and optimization
SPO	smart “predict, then optimize”
QP	quadratic program

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. (2016). Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, CCS ’16, page 308–318, NY, USA. Association for Computing Machinery.

Agrawal, A., Amos, B., Barratt, S., Boyd, S., Diamond, S., and Kolter, J. Z. (2019). Differentiable convex optimization layers. In Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc.

Alley, M., Biggs, M., Hariss, R., Herrmann, C., Li, M. L., and Perakis, G. (2023). Pricing for heterogeneous products: Analytics for ticket reselling. Manufacturing & Service Operations Management, 25(2):409–426.

Amos, B. and Kolter, J. Z. (2017). OptNet: Differentiable optimization as a layer in neural networks. In International Conference on Machine Learning, volume 70, pages 136–145. PMLR.

Aronszajn, N. (1950). Theory of reproducing kernels. Transactions of the American mathematical society, 68(3):337–404.

Athey, S., Tibshirani, J., and Wager, S. (2019). Generalized random forests. The Annals of Statistics, 47(2):1148–1178.

Backhoff, J., Beiglbock, M., Lin, Y., and Zalashko, A. (2017). Causal transport in discrete time and applications. SIAM Journal on Optimization, 27(4):2528–2562.

Bai, S., Kolter, J. Z., and Koltun, V. (2019). Deep equilibrium models. In Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc.

Ban, G.-Y., Gallien, J., and Mersereau, A. J. (2019). Dynamic procurement of new products with covariate information: The residual tree method. Manufacturing & Service Operations Management, 21(4):798–815.

Ban, G.-Y. and Rudin, C. (2019). The Big Data Newsvendor: Practical Insights from Machine Learning. Operations Research, 67(1):90–108.

Basciftci, B., Ahmed, S., and Shen, S. (2021). Distributionally robust facility location problem under decision-dependent stochastic demand. European Journal of Operational Research, 292(2):548–561.

Bazier-Matte, T. and Delage, E. (2020). Generalization bounds for regularized portfolio selection with market side information. INFOR: Information Systems and Operational Research, 58(2):374–401.

Bengio, Y. (1997). Using a financial training criterion rather than a prediction criterion. International Journal of Neural Systems, 8(4):433–443.

Bengio, Y., Lodi, A., and Prouvost, A. (2021). Machine learning for combinatorial optimization: A methodological tour d'horizon. European Journal of Operational Research, 290(2):405–421.

Berthet, Q., Blondel, M., Teboul, O., Cuturi, M., Vert, J.-P., and Bach, F. (2020). Learning with differentiable perturbed optimizers. In Advances in Neural Information Processing Systems, volume 33, pages 9508–9519. Curran Associates, Inc.

Bertsimas, D., Dunn, J., and Mundru, N. (2019). Optimal prescriptive trees. INFORMS Journal on Optimization, 1(2):164–183.

Bertsimas, D. and Kallus, N. (2020). From predictive to prescriptive analytics. Management Science, 66(3):1025–1044.

Bertsimas, D. and Koduri, N. (2022). Data-driven optimization: A reproducing kernel hilbert space approach. Operations Research, 70(1):454–471.

Bertsimas, D., McCord, C., and Sturt, B. (2023). Dynamic optimization with side information. European Journal of Operational Research, 304(2):634–651.

Bertsimas, D. and Van Parys, B. (2022). Bootstrap robust prescriptive analytics. Mathematical Programming, 195(1-2):39–78.

Besbes, O., Ma, W., and Mouchtaki, O. (2023). Quality vs. quantity of data in contextual decision-making: Exact analysis under newsvendor loss. arXiv preprint arXiv:2302.08424.

Birge, J. R. and Louveaux, F. (2011). Introduction to Stochastic Programming. Springer New York, New York, NY.

Blondel, M., Berthet, Q., Cuturi, M., Frostig, R., Hoyer, S., Llinares-López, F., Pedregosa, F., and Vert, J.-P. (2022). Efficient and modular implicit differentiation. In Advances in neural information processing systems, volume 35, pages 5230–5242. Curran Associates, Inc.

Blondel, M., Martins, A. F., and Niculae, V. (2020). Learning with Fenchel-Young losses. Journal of Machine Learning Research, 21(1):1314–1382.

Butler, A. and Kwon, R. H. (2023a). Efficient differentiable quadratic programming layers: an ADMM approach. Computational Optimization and Applications, 84(2):449–476.

Butler, A. and Kwon, R. H. (2023b). Integrating prediction in mean-variance portfolio optimization. Quantitative Finance, 23(3):429–452.

Buttler, S., Philippi, A., Stein, N., and Pibernik, R. (2022). A meta analysis of data-driven newsvendor approaches. In ICLR 2022 Workshop on Setting up ML Evaluation Standards to Accelerate Progress.

Chan, T. C., Mahmood, R., and Zhu, I. Y. (2021). Inverse optimization: Theory and applications. arXiv preprint arXiv:2109.03920.

Chen, B., Donti, P. L., Baker, K., Kolter, J. Z., and Bergés, M. (2021). Enforcing policy feasibility constraints through differentiable projection for energy optimization. In Proceedings of the Twelfth ACM International Conference on Future Energy Systems, page 199–210, Virtual Event Italy. ACM.

Chen, R. and Paschalidis, I. (2019). Selecting optimal decisions via distributionally robust nearest-neighbor regression. In Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc.

Chen, R. and Paschalidis, I. C. (2018). A robust learning approach for regression models based on distributionally robust optimization. Journal of Machine Learning Research, 19(13):1–48.

Chen, W., Tanneau, M., and Van Hentenryck, P. (2023). End-to-end feasible optimization proxies for large-scale economic dispatch. arXiv preprint arXiv:2304.11726.

Chenreddy, A. R., Bandi, N., and Delage, E. (2022). Data-driven conditional robust optimization. In Advances in Neural Information Processing Systems, volume 35, page 9525–9537. Curran Associates, Inc.

Chu, H., Zhang, W., Bai, P., and Chen, Y. (2023). Data-driven optimization for last-mile delivery. Complex & Intelligent Systems, 9(9):2271–2284.

Chung, T.-H., Rostami, V., Bastani, H., and Bastani, O. (2022). Decision-aware learning for optimizing health supply chains. arXiv preprint arXiv:2211.08507.

Ciocan, D. F. and Mišić, V. V. (2022). Interpretable optimal stopping. Management Science, 68(3):1616–1638.

Cohen, M. C., Elmachtoub, A. N., and Lei, X. (2022). Price discrimination with fairness constraints. Management Science, 68(12):8536–8552.

Cohen, M. C., Miao, S., and Wang, Y. (2021). Dynamic pricing with fairness constraints. Available at https://dx.doi.org/10.2139/ssrn.3930622.

Cristian, R., Harsha, P., Perakis, G., Quanz, B. L., and Spantidakis, I. (2022). End-to-end learning via constraint-enforcing approximators for linear programs with applications to supply chains. AI for Decision Optimization Workshop of the AAAI Conference on Artificial Intelligence.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals and Systems, 2(4):303–314.

Dalle, G., Baty, L., Bouvier, L., and Parmentier, A. (2022). Learning with combinatorial optimization layers: a probabilistic approach. arXiv preprint arXiv:2207.13513.

Davis, D. and Yin, W. (2017). A three-operator splitting scheme and its optimization applications. Set-Valued and Variational Analysis, 25(4):829–858.

Demirović, E., J Stuckey, P., Bailey, J., Chan, J., Leckie, C., Ramamohanarao, K., Guns, T., and Kraus, S. (2019). Predict+ optimise with ranking objectives: Exhaustively learning linear functions. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, pages 1078–1085. International Joint Conferences on Artificial Intelligence Organization.

Demirović, E., Stuckey, P. J., Guns, T., Bailey, J., Leckie, C., Ramamohanarao, K., and Chan, J. (2020). Dynamic programming for predict+optimise. Proceedings of the AAAI Conference on Artificial Intelligence, 34(0202):1444–1451.

Deng, Y. and Sen, S. (2022). Predictive stochastic programming. Computational Management Science, 19(1):65–98.

Domke, J. (2012). Generic methods for optimization-based modeling. In International Conference on Artificial Intelligence and Statistics, pages 318–326. PMLR.

Donti, P., Amos, B., and Kolter, J. Z. (2017). Task-based end-to-end model learning in stochastic optimization. In Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc.

Donti, P. L., Roderick, M., Fazlyab, M., and Kolter, J. Z. (2021). Enforcing robust control guarantees within neural network policies. In International Conference on Learning Representations.

Doshi-Velez, F. and Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608.

Duvenaud, D., Kolter, J. Z., and Johnson, M. (2020). Deep implicit layers tutorial — neural ODEs, deep equilibrium models, and beyond. Neural Information Processing Systems Tutorial.

El Balghiti, O., Elmachtoub, A. N., Grigas, P., and Tewari, A. (2019). Generalization bounds in the predict-then-optimize framework. In Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc.

El Balghiti, O., Elmachtoub, A. N., Grigas, P., and Tewari, A. (2022). Generalization bounds in the predict-then-optimize framework. Mathematics of Operations Research. Forthcoming.

Elmachtoub, A. N. and Grigas, P. (2022). Smart "Predict, then Optimize". Management Science, 68(1):9–26.

Elmachtoub, A. N., Lam, H., Zhang, H., and Zhao, Y. (2023). Estimate-then-optimize versus integrated-estimation-optimization: A stochastic dominance perspective. arXiv preprint arXiv:2304.06833.

Elmachtoub, A. N., Liang, J. C. N., and McNellis, R. (2020). Decision trees for decision-making under the predict-then-optimize framework. In International Conference on Machine Learning, pages 2858–2867. PMLR.

Esteban-Pérez, A. and Morales, J. M. (2022). Distributionally robust stochastic programs with side information based on trimmings. Mathematical Programming, 195(1):1069–1105.

Esteban-Pérez, A. and Morales, J. M. (2023). Distributionally robust optimal power flow with contextual information. European Journal of Operational Research, 306(3):1047–1058.

Estes, A. S. and Richard, J.-P. P. (2023). Smart predict-then-optimize for two-stage linear programs with side information. INFORMS Journal on Optimization. Forthcoming.

Fajemisin, A. O., Maragno, D., and den Hertog, D. (2023). Optimization with constraint learning: a framework and survey. European Journal of Operational Research.

Ferber, A., Wilder, B., Dilkina, B., and Tambe, M. (2020). MIPaaL: Mixed Integer Program as a Layer. Proceedings of the AAAI Conference on Artificial Intelligence, 34(02):1504–1511.

Ferreira, K. J., Lee, B. H., and Simchi-Levi, D. (2016). Analytics for an online retailer: Demand forecasting and price optimization. Manufacturing & Service Operations Management, 18(1):69–88.

Forel, A., Parmentier, A., and Vidal, T. (2023). Explainable data-driven optimization: From context to decision and back again. arXiv preprint arXiv:2301.10074.

Fung, S. W., Heaton, H., Li, Q., Mckenzie, D., Osher, S., and Yin, W. (2022). JFB: Jacobian-free backpropagation for implicit networks. Proceedings of the AAAI Conference on Artificial Intelligence, 36(66):6648–6656.

Grefenstette, E., Amos, B., Yarats, D., Htut, P. M., Molchanov, A., Meier, F., Kiela, D., Cho, K., and Chintala, S. (2019). Generalized inner loop meta-learning. arXiv preprint arXiv:1910.01727.

Grigas, P., Qi, M., and Shen, M. (2021). Integrated conditional estimation-optimization. arXiv preprint arXiv:2110.12351.

Hannah, L., Powell, W., and Blei, D. (2010). Nonparametric density estimation for stochastic optimization with an observable state variable. In Advances in Neural Information Processing Systems, volume 23. Curran Associates, Inc.

Hastie, T., Tibshirani, R., and Friedman, J. H. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction, volume 2. Springer.

Heaton, H., McKenzie, D., Li, Q., Fung, S. W., Osher, S., and Yin, W. (2022). Learn to predict equilibria via fixed point networks. arXiv:2106.00906.

Ho-Nguyen, N. and Kılınç-Karzan, F. (2022). Risk guarantees for end-to-end prediction and optimization processes. Management Science, 68(12):8680–8698.

Hofmann, T., Schölkopf, B., and Smola, A. (2008). Kernel methods in machine learning. Annals of Statistics, 36(3):1171–1220.

Hu, Y., Kallus, N., and Mao, X. (2022). Fast rates for contextual linear optimization. Management Science, 68(6):4236–4245.

Huber, J., Müller, S., Fleischmann, M., and Stuckenschmidt, H. (2019). A data-driven newsvendor problem: From data to decision. European Journal of Operational Research, 278(3):904–915.

Ifrim, G., O'Sullivan, B., and Simonis, H. (2012). Properties of energy-price forecasts for scheduling. In International Conference on Principles and Practice of Constraint Programming. Springer.

Jeong, J., Jaggi, P., Butler, A., and Sanner, S. (2022). An exact symbolic reduction of linear smart Predict+Optimize to mixed integer linear programming. In International Conference on Machine Learning, volume 162, pages 10053–10067. PMLR.

Kallus, N. and Mao, X. (2022). Stochastic optimization forests. Management Science, 69(4):1975–1994.

Kaminski, M. E. (2019). The right to explanation, explained. Berkeley Technology Law Journal, 34(1):189–218.

Kannan, R., Bayraksan, G., and Luedtke, J. (2021). Heteroscedasticity-aware residuals-based contextual stochastic optimization. arXiv preprint arXiv:2101.03139.

Kannan, R., Bayraksan, G., and Luedtke, J. R. (2020). Residuals-based distributionally robust optimization with covariate information. arXiv preprint arXiv:2012.01088.

Kannan, R., Bayraksan, G., and Luedtke, J. R. (2022). Data-driven sample average approximation with covariate information. arXiv preprint arXiv:2207.13554.

Kantorovich, L. V. and Rubinshtein, G. S. (1958). On a space of totally additive functions. Vestnik Leningradskogo Universiteta, 13(7):52–59.

Keshavarz, P. (2022). Interpretable contextual newsvendor models: A tree-based method to solving data-driven newsvendor problems. Master's thesis, University of Ottawa.

Kong, L., Cui, J., Zhuang, Y., Feng, R., Prakash, B. A., and Zhang, C. (2022). End-to-end stochastic optimization with energy-based model. In Advances in Neural Information Processing Systems, volume 35, pages 11341–11354. Curran Associates, Inc.

Kotary, J., Dinh, M. H., and Fioretto, F. (2023). Folded optimization for end-to-end model-based learning. In International Joint Conference on Artificial Intelligence.

Kotary, J., Fioretto, F., Van Hentenryck, P., and Wilder, B. (2021). End-to-end constrained optimization learning: A survey. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, page 4475–4482. International Joint Conferences on Artificial Intelligence Organization.

Krantz, S. G. and Parks, H. R. (2002). The Implicit Function Theorem: History, Theory, and Applications. Springer Science & Business Media.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. The Annals of Mathematical Statistics, 22(1):79–86.

Lassalle, R. (2018). Causal transport plans and their Monge–Kantorovich problems. Stochastic Analysis and Applications, 36(3):452–484.

Lawless, C. and Zhou, A. (2022). A note on task-aware loss via reweighing prediction loss by decision-regret. arXiv preprint arXiv:2211.05116.

Lin, S., Chen, Y., Li, Y., and Shen, Z.-J. M. (2022). Data-driven newsvendor problems regularized by a profit risk constraint. Production and Operations Management, 31(4):1630–1644.

Liu, H. and Grigas, P. (2021). Risk bounds and calibration for a smart predict-then-optimize method. In Advances in Neural Information Processing Systems, volume 34, pages 22083–22094. Curran Associates, Inc.

Liu, M., Grigas, P., Liu, H., and Shen, Z.-J. M. (2023a). Active learning in the predict-then-optimize framework: A margin-based approach. arXiv preprint arXiv:2305.06584.

Liu, S., He, L., and Max Shen, Z.-J. (2021). On-time last-mile delivery: Order assignment with travel-time predictors. Management Science, 67(7):4095–4119.

Liu, W. and Zhang, Z. (2023). Solving data-driven newsvendor pricing problems with decision-dependent effect. arXiv preprint arXiv:2304.13924.

Liu, Z., Yin, Y., Bai, F., and Grimm, D. K. (2023b). End-to-end learning of user equilibrium with implicit neural networks. Transportation Research Part C: Emerging Technologies, 150:104085.

Loke, G. G., Tang, Q., and Xiao, Y. (2022). Decision-driven regularization: A blended model for predict-then-optimize. Available at https://dx.doi.org/10.2139/ssrn.3623006.

Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. (2017). The expressive power of neural networks: A view from the width. In Advances in Neural Information Processing Systems.

Mandi, J., Bucarey, V., Tchomba, M. M. K., and Guns, T. (2022). Decision-focused learning: Through the lens of learning to rank. In International Conference on Machine Learning, pages 14935–14947. PMLR.

Mandi, J. and Guns, T. (2020). Interior point solving for LP-based prediction+optimisation. In Advances in Neural Information Processing Systems, volume 33, pages 7272–7282. Curran Associates, Inc.

Mandi, J., Stuckey, P. J., Guns, T., et al. (2020). Smart predict-and-optimize for hard combinatorial optimization problems. Proceedings of the AAAI Conference on Artificial Intelligence, 34(02):1603–1610.

Martínez-de Albeniz, V. and Belkaid, A. (2021). Here comes the sun: Fashion goods retailing under weather fluctuations. European Journal of Operational Research, 294(3):820–830.

Mate, A., Madaan, L., Taneja, A., Madhiwalla, N., Verma, S., Singh, G., Hegde, A., Varakantham, P., and Tambe, M. (2022). Field study in deploying restless multi-armed bandits: Assisting non-profits in improving maternal and child health. Proceedings of the AAAI Conference on Artificial Intelligence, 36(1111):12017–12025.

McKenzie, D., Fung, S. W., and Heaton, H. (2023). Faster predict-and-optimize with three-operator splitting. arXiv preprint arXiv:2301.13395.

Mieth, R., Morales, J. M., and Poor, H. V. (2023). Data valuation from data-driven optimization. arXiv preprint arXiv:2305.01775.

Mišić, V. V. and Perakis, G. (2020). Data analytics in operations management: A review. Manufacturing & Service Operations Management, 22(1):158–169.

Monga, V., Li, Y., and Eldar, Y. C. (2021). Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. IEEE Signal Processing Magazine, 38(2):18–44.

Mulamba, M., Mandi, J., Diligenti, M., Lombardi, M., Lopez, V. B., and Guns, T. (2021). Contrastive losses and solution caching for predict-and-optimize. In 30th International Joint Conference on Artificial Intelligence (IJCAI-21): IJCAI-21, pages 2833–2840. International Joint Conferences on Artificial Intelligence.

Muñoz, M. A., Pineda, S., and Morales, J. M. (2022). A bilevel framework for decision-making under uncertainty with contextual information. Omega, 108:102575.

Nadaraya, E. (1964). On estimating regression. Theory of Probability & its Applications, 9(1):141–142.

Neghab, D. P., Khayyati, S., and Karaesmen, F. (2022). An integrated data-driven method using deep learning for a newsvendor problem with unobservable features. European Journal of Operational Research, 302(2):482–496.

Nguyen, V. A., Zhang, F., Blanchet, J., Delage, E., and Ye, Y. (2020). Distributionally robust local nonparametric conditional estimation. In Advances in Neural Information Processing Systems, volume 33, pages 15232–15242. Curran Associates, Inc.

Nguyen, V. A., Zhang, F., Blanchet, J., Delage, E., and Ye, Y. (2021). Robustifying conditional portfolio decisions via optimal transport. arXiv preprint arXiv:2103.16451.

Nikishin, E., Abachi, R., Agarwal, R., and Bacon, P.-L. (2022). Control-oriented model-based reinforcement learning with implicit differentiation. Proceedings of the AAAI Conference on Artificial Intelligence, 36(7):7886–7894.

Notz, P. M. and Pibernik, R. (2022). Prescriptive analytics for flexible capacity management. Management Science, 68(3):1756–1775.

Ohmori, S. (2021). A predictive prescription using minimum volume k-nearest neighbor enclosing ellipsoid and robust optimization. Mathematics, 9(2):119.

Oroojlooyjadid, A., Snyder, L. V., and Takáč, M. (2020). Applying deep learning to the newsvendor problem. IISE Transactions, 52(4):444–463.

Perakis, G., Sim, M., Tang, Q., and Xiong, P. (2023). Robust pricing and production with information partitioning and adaptation. Management Science, 69(3):1398–1419.

Perrault, A., Wilder, B., Ewing, E., Mate, A., Dilkina, B., and Tambe, M. (2020). End-to-End Game-Focused Learning of Adversary Behavior in Security Games. Proceedings of the AAAI Conference on Artificial Intelligence, 34(02):1378–1386.

Peršak, E. and Anjos, M. F. (2023). Contextual robust optimisation with uncertainty quantification. In Proceedings of the 20th International Conference on the Integrationg of Constraint Programming, Artificial Intelligence, and Operations Research, pages 124–132. Springer.

Pineda, L., Fan, T., Monge, M., Venkataraman, S., Sodhi, P., Chen, R. T., Ortiz, J., DeTone, D., Wang, A., Anderson, S., et al. (2022). Theseus: A library for differentiable nonlinear optimization. In Advances in Neural Information Processing Systems, volume 35, pages 3801–3818. Curran Associates, Inc.

Qi, M. and Shen, Z.-J. (2022). Integrating prediction/estimation and optimization with applications in operations management. In Tutorials in Operations Research: Emerging and Impactful Topics in Operations, pages 36–58. INFORMS.

Qi, M., Shi, Y., Qi, Y., Ma, C., Yuan, R., Wu, D., and Shen, Z.-J. M. (2023). A practical end-to-end inventory management model with deep learning. Management Science, 69(2):759–773.

Rahimian, H. and Pagnoncelli, B. (2022). Data-driven approximation of contextual chance-constrained stochastic programs. Available at https://optimization-online.org/?p=20569.

Ren, J., Feng, X., Liu, B., Pan, X., Fu, Y., Mai, L., and Yang, Y. (2022). Torchopt: An efficient library for differentiable optimization. arXiv preprint arXiv:2211.06934.

Rios, I., Wets, R. J., and Woodruff, D. L. (2015). Multi-period forecasting and scenario generation with limited data. Computational Management Science, 12(2):267–295.

Rockafellar, R. T. and Wets, R. J.-B. (2009). Variational Analysis. Springer, Berlin.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence, 1(5):206–215.

Rust, J. (1988). Maximum likelihood estimation of discrete control processes. SIAM Journal on Control and Optimization, 26(5):1006–1024.

Rychener, Y. and Sutter, D. K. T. (2023). End-to-end learning for stochastic optimization: A bayesian perspective. In International Conference on Machine Learning.

Sang, L., Xu, Y., Long, H., Hu, Q., and Sun, H. (2022). Electricity price prediction for energy storage system arbitrage: A decision-focused approach. IEEE Transactions on Smart Grid, 13(4):2822–2832.

Sang, L., Xu, Y., Long, H., and Wu, W. (2023). Safety-aware semi-end-to-end coordinated decision model for voltage regulation in active distribution network. IEEE Transactions on Smart Grid, 14(3):1814–1826.

Sen, S. and Deng, Y. (2017). Learning enabled optimization: Towards a fusion of statistical learning and stochastic programming. Available at https://optimization-online.org/?p=14456.

Serrano, B., Minner, S., Schiffer, M., and Vidal, T. (2022). Bilevel optimization for feature selection in the data-driven newsvendor problem. arXiv preprint arXiv:2209.05093.

Shah, S., Wang, K., Wilder, B., Perrault, A., and Tambe, M. (2022). Decision-focused learning without decision-making: Learning locally optimized decision losses. In Advances in Neural Information Processing Systems, volume 35, pages 1320–1332. Curran Associates, Inc.

Shapiro, A., Dentcheva, D., and Ruszczyński, A. (2014). Lectures on Stochastic Programming: Modeling and Theory. SIAM, Philadelphia, second edition.

Shlezinger, N., Eldar, Y. C., and Boyd, S. P. (2022). Model-based deep learning: On the intersection of deep learning and optimization. IEEE Access, 10:115384–115398.

Simonis, H., O'Sullivan, B., Mehta, D., Hurley, B., and Cauwer, M. D. (2014). CSPLib problem 059: Energy-cost aware scheduling. http://www.csplib.org/Problems/prob059. Accessed on June 14, 2023.

Smith, J. E. and Winkler, R. L. (2006). The optimizer's curse: Skepticism and postdecision surprise in decision analysis. Management Science, 52(3):311–322.

Srivastava, P. R., Wang, Y., Hanasusanto, G. A., and Ho, C. P. (2021). On data-driven prescriptive analytics with side information: A regularized Nadaraya-Watson approach. arXiv preprint arXiv:2110.04855.

Stratigakos, A., Camal, S., Michiorri, A., and Kariniotakis, G. (2022). Prescriptive trees for integrated forecasting and optimization applied in trading of renewable energy. IEEE Transactions on Power Systems, 37(6):4696–4708.

Sun, C., Liu, S., and Li, X. (2023a). Maximum optimality margin: A unified approach for contextual linear programming and inverse linear programming. arXiv preprint arXiv:2301.11260.

Sun, H., Shi, Y., Wang, J., Tuan, H. D., Poor, H. V., and Tao, D. (2023b). Alternating differentiation for optimization layers. In International Conference on Learning Representations.

Sun, J., Li, H., Xu, Z., et al. (2016). Deep ADMM-Net for compressive sensing MRI. In Advances in Neural Information Processing Systems, volume 29. Curran Associates, Inc.

Sun, X., Leung, C. H., Li, Y., and Wu, Q. (2023c). A unified perspective on regularization and perturbation in differentiable subset selection. In International Conference on Artificial Intelligence and Statistics, pages 4629–4642. PMLR.

Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. In Advances in Neural Information Processing Systems, volume 12. MIT Press.

Tang, B. and Khalil, E. B. (2022). PyEPO: A PyTorch-based end-to-end predict-then-optimize library for linear and integer programming. arXiv preprint arXiv:2206.14234.

Tian, X., Yan, R., Liu, Y., and Wang, S. (2023). A smart predict-then-optimize method for targeted and cost-effective maritime transportation. Transportation Research Part B: Methodological, 172:32–52.

Van Parys, B. P., Esfahani, P. M., and Kuhn, D. (2021). From data to decisions: Distributionally robust optimization is optimal. Management Science, 67(6):3387–3402.

Vlastelica, M., Paulus, A., Musil, V., Martius, G., and Rolinek, M. (2019). Differentiation of blackbox combinatorial solvers. In International Conference on Learning Representations.

Vohra, R., Rajaei, A., and Cremer, J. L. (2023). End-to-end learning with multiple modalities for system-optimised renewables nowcasting. arXiv preprint arXiv:2304.07151.

Wager, S. and Athey, S. (2018). Estimation and inference of heterogeneous treatment effects using random forests. Journal of the American Statistical Association, 113(523):1228–1242.

Wahdany, D., Schmitt, C., and Cremer, J. L. (2023). More than accuracy: end-to-end wind power forecasting that optimises the energy system. Electric Power Systems Research, 221:109384.

Wang, K., Verma, S., Mate, A., Shah, S., Taneja, A., Madhiwalla, N., Hegde, A., and Tambe, M. (2023). Scalable decision-focused learning in restless multi-armed bandits with application to maternal and child health. arXiv preprint arXiv:2202.00916.

Wang, K., Wilder, B., Perrault, A., and Tambe, M. (2020). Automatically learning compact quality-aware surrogates for optimization problems. In Advances in Neural Information Processing Systems, volume 33, pages 9586–9596. Curran Associates, Inc.

Wang, T., Chen, N., and Wang, C. (2021). Distributionally robust prescriptive analytics with Wasserstein distance. arXiv preprint arXiv:2106.05724.

Watson, G. (1964). Smooth regression analysis. Sankhyā: The Indian Journal of Statistics, Series A, 26(4):359–372.

Wilder, B., Dilkina, B., and Tambe, M. (2019a). Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. Proceedings of the AAAI Conference on Artificial Intelligence, 33(01):1658–1665.

Wilder, B., Ewing, E., Dilkina, B., and Tambe, M. (2019b). End to end learning and optimization on graphs. In Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc.

Xie, X., Wu, J., Liu, G., Zhong, Z., and Lin, Z. (2019). Differentiable linearized admm. In International Conference on Machine Learning, pages 6902–6911. PMLR.

Xie, Y., Dai, H., Chen, M., Dai, B., Zhao, T., Zha, H., Wei, W., and Pfister, T. (2020). Differentiable top-k with optimal transport. In Advances in Neural Information Processing Systems, volume 33, page 20520–20531. Curran Associates, Inc.

Xu, Y. and Cohen, S. B. (2018). Stock movement prediction from tweets and historical prices. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1970–1979. Association for Computational Linguistics.

Yan, R., Wang, S., and Fagerholt, K. (2020). A semi-"smart predict then optimize" (semi-SPO) method for efficient ship inspection. Transportation Research Part B: Methodological, 142:100–125.

Yang, J., Zhang, L., Chen, N., Gao, R., and Hu, M. (2023). Decision-making with side information: A causal transport robust approach. Available at https://optimization-online.org/?p=20639.

Yanıkoğlu, İ., Gorissen, B. L., and den Hertog, D. (2019). A survey of adjustable robust optimization. European Journal of Operational Research, 277(3):799–813.

Zhang, C., Zhang, Z., Cucuringu, M., and Zohren, S. (2021). A universal end-to-end approach to portfolio optimization via deep learning. arXiv preprint arXiv:2111.09170.

Zhang, L., Yang, J., and Gao, R. (2023a). Optimal robust policy for feature-based newsvendor. Management Science. Forthcoming.

Zhang, Y. and Gao, J. (2017). Assessing the performance of deep learning algorithms for newsvendor problem. In Liu, D., Xie, S., Li, Y., Zhao, D., and El-Alfy, E.-S. M., editors, Neural Information Processing, pages 912–921, Cham. Springer International Publishing.

Zhang, Y., Liu, J., and Zhao, X. (2023b). Data-driven piecewise affine decision rules for stochastic programming with covariate information. arXiv preprint arXiv:2304.13646.

Zhu, T., Xie, J., and Sim, M. (2022). Joint estimation and robustness optimization. Management Science, 68(3):1659–1677.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., and Dey, A. K. (2008). Maximum entropy inverse reinforcement learning. Proceedings of the AAAI Conference on Artificial Intelligence, 8:1433–1438.