Learning Multi-Objective Program Through Online Learning

Chaosheng Dong CHAOSD@AMAZON.COM
Yijia Wang YIW94@PITT.EDU
Bo Zeng BZENG@PITT.EDU

Abstract

We investigate the problem of learning the parameters (i.e., objective functions or constraints) of a multi-objective decision making model, based on a set of sequentially arrived decisions. In particular, these decisions might not be exact and possibly carry measurement noise or are generated with the bounded rationality of decision makers. In this paper, we propose a general online learning framework to deal with this learning problem using inverse multi-objective optimization, and prove that this framework converges at a rate of $\mathcal{O}(1/\sqrt{T})$ under certain regularity conditions. More precisely, we develop two online learning algorithms with implicit update rules which can handle noisy data. Numerical results with both synthetic and real world datasets show that both algorithms can learn the parameters of a multi-objective program with great accuracy and are robust to noise.

1. Introduction

In this paper, we aim to learn the parameters (i.e., constraints and a set of objective functions) of a decision making problem with multiple objectives, instead of solving for its efficient (or Pareto) optimal solutions, which is the typical scenario. More precisely, we seek to learn θ given $\{\mathbf{y}_i\}_{i\in[N]}$ that are observations of the efficient solutions of the multi-objective optimization problem (MOP):

$$\begin{array}{ll} \min\limits_{\mathbf{x}} & \{f_1(\mathbf{x},\theta), f_2(\mathbf{x},\theta), \ldots, f_p(\mathbf{x},\theta)\} \\ s.t. & \mathbf{x} \in X(\theta), \end{array}$$

where θ is the true but unknown parameter of the MOP. In particular, we consider such learning problems in online fashion, noting observations are unveiled sequentially in practical scenarios. Specifically, we study such learning problem as an inverse multi-objective optimization problem (IMOP) dealing with noisy data, develop online learning algorithms to derive parameters for each objective function and constraint, and output an estimation of the distribution of weights (which, together with objective functions, define individuals' utility functions) among human subjects.

Learning human participants' decision making scheme is critical for an organization in designing and providing services or products. Nevertheless, as in most scenarios, we can only observe their decisions or behaviors and cannot directly access decision making schemes. Indeed, participants probably do not have exact information regarding their own decision making process [15]. To bridge the discrepancy, we leverage the inverse optimization idea that has been proposed and received significant attention in the optimization community, which is to infer the missing information of the underlying decision models from observed data, assuming that human decision makers are making optimal decisions [1, 3, 4, 6, 10, 14, 15, 19–21]. This subject actually carries the data-driven concept and becomes more applicable as large amounts of data are generated and become readily available, especially those from digital devices and online transactions.

Our work is most related to the subject of inverse multi-objective optimization. The goal is to find multiple objective functions that explain the observed efficient solutions well. There are several recent studies related to the presented research. One is in Chan et al. [6], which considers a single observation that is assumed to be an exact optimal solution. Then, given a set of well-defined linear functions, an inverse optimization is formulated to learn their weights. Another one is Dong and Zeng [8], which proposes the batch learning framework to infer utility functions or constraints from multiple noisy decisions through inverse multi-objective optimization. This work can be categorized as doing inverse multi-objective optimization in batch setting. Recently, Dong and Zeng [9] extends Dong and Zeng [8] with distributionally robust optimization by leveraging the prominent Wasserstein metric. In contrast, we do inverse multi-objective optimization in online settings, and the proposed online learning algorithms significantly accelerate the learning process with performance guarantees, allowing us to deal with more realistic and complex preference inference problems.

To the best of authors' knowledge, we propose the first general framework of online learning for inferring decision makers' objective functions or constraints using inverse multi-objective optimization. This framework can learn the parameters of any convex decision making problem, and can explicitly handle noisy decisions. Moreover, we show that the online learning approach, which adopts an implicit update rule, has an $\mathcal{O}(\sqrt{T})$ regret under suitable regularity conditions when using the ideal loss function. We finally illustrate the performance of two algorithms on both a multi-objective quadratic programming problem and a portfolio optimization problem. Results show that both algorithms can learn parameters with great accuracy and are robust to noise while the second algorithm significantly accelerate the learning process over the first one.

2. Problem setting

2.1. Decision making problem with multiple objectives

We consider a family of parametrized multi-objective decision making problems of the form

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathbb{R}^n} & \left\{f_1(\mathbf{x},\theta), f_2(\mathbf{x},\theta),\ldots,f_p(\mathbf{x},\theta)\right\} \\ s.t. & \mathbf{x}\in X(\theta), \end{array} \tag{DMP}$$

where $p \geq 2$ and $f_l(\mathbf{x},\theta) : \mathbb{R}^n \times \mathbb{R}^{n_\theta} \mapsto \mathbb{R}$ for each $l \in [p]$. Assume parameter $\theta \in \Theta \subseteq \mathbb{R}^{n_\theta}$. We denote the vector of objective functions by $\mathbf{f}(\mathbf{x},\theta) = (f_1(\mathbf{x},\theta), f_2(\mathbf{x},\theta),\ldots,f_p(\mathbf{x},\theta))^T$. Assume $X(\theta) = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{g}(\mathbf{x},\theta) \leq \mathbf{0}, \mathbf{x} \in \mathbb{R}^n_+\}$, where $\mathbf{g}(\mathbf{x},\theta) = (g_1(\mathbf{x},\theta),\ldots,g_q(\mathbf{x},\theta))^T$ is another vector-valued function with $g_k(\mathbf{x},\theta) : \mathbb{R}^n \times \mathbb{R}^{n_\theta} \mapsto \mathbb{R}$ for each $k \in [q]$.

Definition 1 (Efficiency) *For fixed θ, a decision vector $\mathbf{x}^* \in X(\theta)$ is said to be efficient if there exists no other decision vector $\mathbf{x} \in X(\theta)$ such that $f_i(\mathbf{x},\theta) \leq f_i(\mathbf{x}^*,\theta)$ for all $i \in [p]$, and $f_k(\mathbf{x},\theta) < f_k(\mathbf{x}^*,\theta)$ for at least one $k \in [p]$.*

In the study of multi-objective optimization, the set of all efficient solutions is denoted by $X_E(\theta)$ and called the efficient set. The weighting method is commonly used to obtain an efficient solution through computing the problem of weighted sum (PWS) Gass and Saaty [11] as follows.

$$\begin{array}{ll} \min & w^T\mathbf{f}(\mathbf{x},\theta) \\ s.t. & \mathbf{x}\in X(\theta), \end{array} \tag{PWS}$$

where $w = (w^1, \ldots, w^p)^T$. Without loss of generality, all possible weights are restricted to a simplex, which is denoted by $\mathscr{W}_p = \{w \in \mathbb{R}^p_+ : \ \mathbf{1}^T w = 1\}$. Next, we denote the set of optimal solutions for the (PWS) by

$$S(w, \theta) = \arg\min_{\mathbf{x}} \left\{ w^T \mathbf{f}(\mathbf{x}, \theta) : \mathbf{x} \in X(\theta) \right\}.$$

Let $\mathscr{W}_p^+ = \{w \in \mathbb{R}^p_{++} : \ \mathbf{1}^T w = 1\}$. Following from Theorem 3.1.2 of Miettinen [18], we have:

Proposition 2 *If $\mathbf{x} \in S(w, \theta)$ and $w \in \mathscr{W}_p^+$, then $\mathbf{x} \in X_E(\theta)$.*

The next result from Theorem 3.1.4 of Miettinen [18] states that all the efficient solutions can be found by the weighting method for convex MOP.

Proposition 3 *Assume that MOP is convex. If $\mathbf{x} \in X$ is an efficient solution, then there exists a weighting vector $w \in \mathscr{W}_p$ such that $\mathbf{x}$ is an optimal solution of (PWS).*

By Propositions 2 - 3, we summarize the relationship between $S(w, \theta)$ and $X_E(\theta)$ as follows.

Corollary 4 *For convex MOP,*

$$\bigcup_{w \in \mathscr{W}_p^+} S(w, \theta) \subseteq X_E(\theta) \subseteq \bigcup_{w \in \mathscr{W}_p} S(w, \theta).$$

2.2. Inverse multi-objective optimization

We denote $\mathbf{y}$ the observed noisy decision that might carry measurement error or is generated with a bounded rationality of the decision maker. We emphasize that this noisy setting of $\mathbf{y}$ reflects the real world situation rather than for analysis of regret, where the noises might be from multiple sources. Throughout the paper we assume that $\mathbf{y}$ is a random variable distributed according to an unknown distribution $\mathbb{P}_{\mathbf{y}}$ supported on $\mathcal{Y}$. As $\mathbf{y}$ is a noisy observation, we note that $\mathbf{y}$ does not necessarily belong to $X(\theta)$, i.e., it might be either feasible or infeasible with respect to $X(\theta)$.

We next discuss the construction of an appropriate loss function for the inverse multi-objective optimization problem [8, 9]. Ideally, given a noisy decision $\mathbf{y}$ and a hypothesis θ, the loss function can be defined as the minimum distance between $\mathbf{y}$ and the efficient set $X_E(\theta)$:

$$l(\mathbf{y}, \theta) = \min_{\mathbf{x} \in X_E(\theta)} \|\mathbf{y} - \mathbf{x}\|_2^2. \qquad \text{(loss function)}$$

For a general MOP, however, there might exist no explicit way to characterize the efficient set $X_E(\theta)$. Hence, an approximation approach to practically describe this is adopted. Following from Corollary 4, a sampling approach is adopted to generate $w_k \in \mathscr{W}_p$ for each $k \in [K]$ and approximate $X_E(\theta)$ as $\bigcup_{k \in [K]} S(w_k, \theta)$. Then, the *surrogate loss function* is defined as

$$l_K(\mathbf{y}, \theta) = \min_{\mathbf{x} \in \bigcup\limits_{k \in [K]} S(w_k, \theta)} \|\mathbf{y} - \mathbf{x}\|_2^2. \qquad \text{(surrogate loss)}$$

By using binary variables, this surrogate loss can be converted into the *Surrogate Loss Problem*.

$$\begin{aligned} l_K(\mathbf{y}, \theta) \quad = \quad & \min_{z_j \in \{0,1\}} \|\mathbf{y} - \sum_{k \in [K]} z_k \mathbf{x}_k\|_2^2 \\ \text{s.t.} \quad & \sum_{k \in [K]} z_k = 1, \ \mathbf{x}_k \in S(w_k, \theta). \end{aligned} \qquad (1)$$

Constraint $\sum_{k \in [K]} z_k = 1$ ensures that exactly one of the efficient solutions will be chosen to measure the distance to $\mathbf{y}$. Hence, solving this optimization problem identifies some w_k with $k \in [K]$ such that the corresponding efficient solution $S(w_k, \theta)$ is closest to $\mathbf{y}$.

3. Online learning for IMOP

In our online learning setting, noisy decisions become available to the learner one by one. Hence, the learning algorithm produces a sequence of hypotheses $(\theta_1, \ldots, \theta_{T+1})$. Here, T is the total number of rounds, and θ_1 is an arbitrary initial hypothesis and θ_t for $t > 1$ is the hypothesis chosen after seeing the $(t-1)$th decision. Let $l(\mathbf{y}_t, \theta_t)$ denote the loss the learning algorithm suffers when it tries to predict $\mathbf{y}_t$ based on the previous observed decisions $\{\mathbf{y}_1, \ldots, \mathbf{y}_{t-1}\}$. The goal of the learner is to minimize the regret, which is the cumulative loss $\sum_{t=1}^{T} l(\mathbf{y}_t, \theta_t)$ against the best possible loss when the whole batch of decisions are available. Formally, the regret is defined as

$$R_T = \sum_{t=1}^{T} l(\mathbf{y}_t, \theta_t) - \min_{\theta \in \Theta} \sum_{t=1}^{T} l(\mathbf{y}_t, \theta).$$

3.1. Online implicit updates

Once receiving the tth noisy decision $\mathbf{y}_t$, the ideal way to update θ_{t+1} is by solving the following optimization problem using the ideal loss function:

$$\theta_{t+1} = \arg\min_{\theta \in \Theta} \frac{1}{2} \|\theta - \theta_t\|_2^2 + \eta_t l(\mathbf{y}_t, \theta), \tag{2}$$

where η_t is the learning rate in each round, and $l(\mathbf{y}_t, \theta)$ is defined in loss function.

As explained in the previous section, $l(\mathbf{y}_t, \theta)$ might not be computable due to the non-existence of the closed form of the efficient set $X_E(\theta)$. Thus, we seek to approximate the update 2 by:

$$\theta_{t+1} = \arg\min_{\theta \in \Theta} \frac{1}{2} \|\theta - \theta_t\|_2^2 + \eta_t l_K(\mathbf{y}_t, \theta), \tag{3}$$

where η_t is the learning rate in each round, and $l_K(\mathbf{y}_t, \theta)$ is defined in surrogate loss.

To solve 3, we can replace $\mathbf{x}_k \in S(w_k, \theta)$ by KKT conditions for each $k \in [K]$:

$$\begin{array}{lll}
\min\limits_{\theta} & \frac{1}{2}\|\theta - \theta_t\|_2^2 + \eta_t \sum\limits_{k \in [K]} \|\mathbf{y}_t - \vartheta_k\|_2^2 & \\
\text{s.t.} & \theta \in \Theta, & \\
& \left[\begin{array}{l} \mathbf{g}(\mathbf{x}_k) \leq \mathbf{0}, \quad \mathbf{u}_k \geq \mathbf{0}, \\ \mathbf{u}_k^T \mathbf{g}(\mathbf{x}_k) = 0, \\ \nabla_{\mathbf{x}_k} w_k^T \mathbf{f}(\mathbf{x}_k, \theta) + \mathbf{u}_k \cdot \nabla_{\mathbf{x}_k} \mathbf{g}(\mathbf{x}_k) = \mathbf{0}, \end{array} \right], & \forall k \in [K], \\
& \mathbf{0} \leq \vartheta_k \leq M_k z_k \mathbf{1}_n, & \forall k \in [K], \\
& \mathbf{x}_k - M_k(1 - z_k)\mathbf{1}_n \leq \vartheta_k \leq \mathbf{x}_k, & \forall k \in [K], \\
& \sum\limits_{k \in [K]} z_k = 1, & \\
& \mathbf{x}_k \in \mathbb{R}^n, \quad \mathbf{u}_k \in \mathbb{R}_+^m, \quad z_k \in \{0, 1\}, & \forall k \in [K],
\end{array}$$

where $\mathbf{u}_k$ is the dual variable for $g_k(\mathbf{x}, \theta) \leq 0$, and M_k is a big number to linearize $z_k \mathbf{x}_k$ [2].

Alternatively, solving 3 is equivalent to solving K independent programs defined in the following and taking the one with the least optimal value (breaking ties arbitrarily).

$$\begin{array}{ll}
\min\limits_{\theta \in \Theta} & \frac{1}{2}\|\theta - \theta_t\|_2^2 + \eta_t \|\mathbf{y}_t - \mathbf{x}\|_2^2 \\
\text{s.t.} & \mathbf{x} \in S(w_k, \theta).
\end{array} \tag{4}$$

Algorithm 1 Online Learning for IMOP

```
Input: noisy decisions {y_t}_{t∈T}, weights {w_k}_{k∈K}
Initialize θ_1 = 0
for t = 1 to T do
    receive y_t
    suffer loss l_K(y_t, θ_t)
    if l_K(y_t, θ_t) = 0 then
        θ_{t+1} ← θ_t
    else
        set learning rate η_t ∝ 1/√t
        update θ_{t+1} by solving 3 directly (or equivalently solving K subproblems 4)
    end if
end for
```

Algorithm 2 Accelerated Online Learning

```
Input: {y_t}_{t∈T} and {w_k}_{k∈K}
Initialize θ_1 = 0
for t = 1 to T do
    receive y_t
    suffer loss l_K(y_t, θ_t)
    let k* = arg min_{k∈[K]} ||y_t − x_k||_2^2, where x_k ∈ S(w_k, θ_t) for k ∈ [K]
    if l_K(y_t, θ_t) = 0 then
        θ_{t+1} ← θ_t
    else
        set learning rate η_t ∝ 1/√t
        update θ_{t+1} by 4 with k = k*
    end if
end for
```

Our application of the implicit update rule to learn an MOP is outlined in Algorithm 1.

Acceleration of Algorithm 1: Note that we update θ and the weight sample assigned to $\mathbf{y}_t$ in 3 simultaneously, meaning both θ and the weight sample index k are variables when solving 3. In other words, one needs to solve K subproblems 4 to get an optimal solution for 3. However, note that the increment of θ by 3 is typically small for each update. Consequently, the weight sample assigned to $\mathbf{y}_t$ using θ_{t+1} is roughly the same as using the previous guess of this parameter, i.e., θ_t. Hence, it is reasonable to approximate 3 by first assigning a weight sample to $\mathbf{y}_t$ based on the previous updating result. Then, instead of computing K problems of 4, we simply compute a single one associated with the selected weight samples, which significantly eases the burden of solving 3. Our application of the accelerated implicit update rule proceeds as outlined in Algorithm 2.

3.2. Analysis of convergence

Note that the proposed online learning algorithms are generally applicable to learn the parameter of any convex MOP. In this section, we show that the average regret converges at a rate of $\mathcal{O}(1/\sqrt{T})$ under certain regularity conditions based on the ideal loss function $l(\mathbf{y}, \theta)$. Namely, we consider the regret bound when using the ideally implicit update rule 2.

Theorem 5 *Suppose Assumptions A.2 - A.4 hold. Then, choosing* $\eta_t = \frac{D\lambda}{2\sqrt{2}(B+R)\kappa}\frac{1}{\sqrt{t}}$, *we have*

$$R_T \leq \frac{4\sqrt{2}(B+R)D\kappa}{\lambda}\sqrt{T}.$$

4. Experiments

In this section, we will provide a multi-objective quadratic program (MQP) and a portfolio optimization problem (see Appendix A.7) to illustrate the performance of the proposed online learning Algorithms 1 and 2. The mixed integer second order conic problems (MISOCPs), which are derived from using KKT conditions in 3, are solved by [12]. Consider the following multi-objective quadratic optimization problem.

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathbb{R}^2_+} & \begin{pmatrix} f_1(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q_1 \mathbf{x} + \mathbf{c}_1^T\mathbf{x} \\ f_2(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q_2 \mathbf{x} + \mathbf{c}_2^T\mathbf{x} \end{pmatrix} \\ s.t. & A\mathbf{x} \leq \mathbf{b}, \end{array}$$

where parameters of the objective functions and constraints are provided in Appendix.

Suppose there are T decision makers. In each round, the learner would receive one noisy decision. Her goal is to learn the objective functions or restrictions of these decision makers. In round t, we suppose that the decision maker derives an efficient solution $\mathbf{x}_t$ by solving (PWS) with weight w_t, which is uniformly chosen from $\mathscr{W}_2$. Next, the learner receives the noisy decision $\mathbf{y}_t$ corrupted by noise that has a jointly uniform distribution with support $[-0.5, 0.5]^2$. Namely, $\mathbf{y}_t = \mathbf{x}_t + \epsilon_t$, where each element of $\epsilon_t \sim U(-0.5, 0.5)$. The learner seeks to learn $\mathbf{c}_1$ and $\mathbf{c}_2$. The learning rate is set to $\eta_t = 5/\sqrt{t}$. Then, we implement Algorithms 1 and 2. At each round t, we solve 4 using parallel computing with 6 workers.

To illustrate the performance of the algorithms in a statistical way, we run 100 repetitions of the experiments. Figure 1 (a) shows the total estimation errors of $\mathbf{c}_1$ and $\mathbf{c}_2$ in each round over the 100 repetitions for the two algorithms. We also plot the average estimation error of the 100 repetitions. As can be seen in this figure, convergence for both algorithms is pretty fast. Also, estimation errors over rounds for different repetitions concentrate around the average, indicating that our algorithm is pretty robust to noise. The estimation error in the last round is not zero because we use a finite K to approximate the efficient set. We see in Figure 1 (b) that Algorithm 2 is much faster than Algorithm 1 especially when K is large. To further illustrate the performance of algorithms, we randomly pick one repetition using Algorithm 1 and plot the estimated efficient set in Figure 1 (c). We can see clearly that the estimated efficient set almost coincides with the real efficient set. Moreover, Figure 1 (d) shows that IMOP in online settings is drastically faster than in batch setting. It is practically impossible to apply the batch setting algorithms in real-world applications.

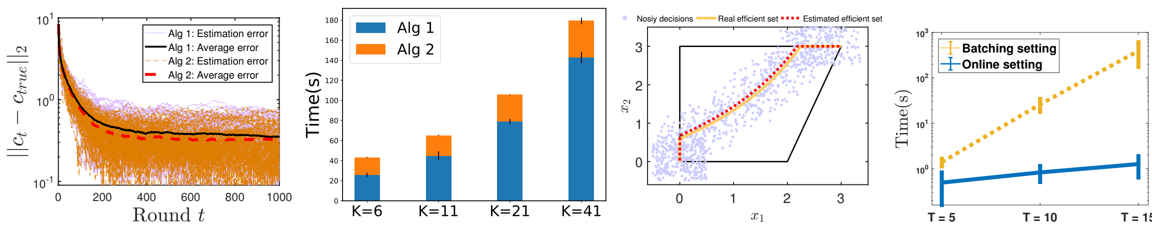

Figure 1: (a) We plot estimation errors at each round t for all 100 experiments and their average estimation errors with $K = 41$. (b) Blue and yellow bars indicate average running time and standard deviations for each K using Algorithm 1 and 2, respectively. (c) We randomly pick one repetition. The estimated efficient set after $T = 1000$ rounds is indicated by the red line. The real efficient set is shown by the yellow line. (d) The dotted brown line is the error bar plot of the running time over 10 repetitions in batch setting. The blue line is the error bar plot of the running time over 100 repetitions in an online setting using Algorithm 1.

References

[1] Ravindra K Ahuja and James B Orlin. Inverse optimization. *Operations Research*, 49(5): 771–783, 2001.

[2] Mohammad Asghari, Amir M Fathollahi-Fard, SMJ Mirzapour Al-e hashem, and Maxim A Dulebenets. Transformation and linearization techniques in optimization: A state-of-the-art survey. *Mathematics*, 2022.

[3] Anil Aswani, Zuo-Jun Shen, and Auyon Siddiq. Inverse optimization with noisy data. *Operations Research*, 2018.

[4] Dimitris Bertsimas, Vishal Gupta, and Ioannis Ch Paschalidis. Data-driven estimation in equilibrium using inverse optimization. *Mathematical Programming*, 153(2):595–633, 2015.

[5] J Frédéric Bonnans and Alexander Shapiro. Optimization problems with perturbations: A guided tour. *SIAM Review*, 40(2):228–264, 1998.

[6] Timothy CY Chan, Tim Craig, Taewoo Lee, and Michael B Sharpe. Generalized inverse multiobjective optimization with application to cancer therapy. *Operations Research*, 62(3): 680–695, 2014.

[7] Chaosheng Dong and Bo Zeng. Inferring parameters through inverse multiobjective optimization. *arXiv preprint arXiv:1808.00935*, 2018.

[8] Chaosheng Dong and Bo Zeng. Expert learning through generalized inverse multiobjective optimization: Models, insights, and algorithms. In *ICML*, 2020.

[9] Chaosheng Dong and Bo Zeng. Wasserstein distributionally robust inverse multiobjective optimization. In *AAAI*, 2021.

[10] Peyman Mohajerin Esfahani, Soroosh Shafieezadeh-Abadeh, Grani A Hanasusanto, and Daniel Kuhn. Data-driven inverse optimization with imperfect information. *Mathematical Programming*, 167(1):191–234, 2018.

[11] Saul Gass and Thomas Saaty. The computational algorithm for the parametric objective function. *Naval Research Logistics*, 2(1-2):39–45, 1955.

[12] Inc. Gurobi Optimization. Gurobi optimizer reference manual, 2016. URL `http://www.gurobi.com`.

[13] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[14] Garud Iyengar and Wanmo Kang. Inverse conic programming with applications. *Operations Research Letters*, 33(3):319–330, 2005.

[15] Arezou Keshavarz, Yang Wang, and Stephen Boyd. Imputing a convex objective function. In *Intelligent Control (ISIC), 2011 IEEE International Symposium on*, pages 613–619. IEEE, 2011.

[16] Brian Kulis and Peter L Bartlett. Implicit online learning. In *ICML*, 2010.

[17] Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.

[18] Kaisa Miettinen. *Nonlinear Multiobjective Optimization*, volume 12. Springer Science & Business Media, 2012.

[19] Andrew J. Schaefer. Inverse integer programming. *Optimization Letters*, 3(4):483–489, 2009.

[20] Yingcong Tan, Daria Terekhov, and Andrew Delong. Learning linear programs from optimal decisions. *arXiv preprint arXiv:2006.08923*, 2020.

[21] Lizhi Wang. Cutting plane algorithms for the inverse mixed integer linear programming problem. *Operations Research Letters*, 37(2):114–116, 2009.

Appendix A. Appendix

A.1. Omitted mathematical reformulations

Before giving the reformulations, we first make some discussions about the surrogate loss functions.

$$\begin{aligned} l_K(\mathbf{y},\theta) &= \min_{z_k\in\{0,1\}} \|\mathbf{y} - \sum_{k\in[K]} z_k\mathbf{x}_k\|_2^2 \\ &= \min_{z_k\in\{0,1\}} \sum_{k\in[K]} \|\mathbf{y} - z_k\mathbf{x}_k\|_2^2 - (K-1)\|\mathbf{y}\|_2^2 \end{aligned}$$

where $\mathbf{x}_k \in S(w_k,\theta)$ and $\sum_{k\in[K]} z_k = 1$.

Since $(K-1)\|\mathbf{y}\|_2^2$ is a constant, we can safely drop it and use the following as the surrogate loss function when solving the optimization program in the implicit update,

$$l_K(\mathbf{y},\theta) = \min_{z_k\in\{0,1\}} \sum_{k\in[K]} \|\mathbf{y} - z_k\mathbf{x}_k\|_2^2$$

where $\mathbf{x}_k \in S(w_k,\theta)$ and $\sum_{k\in[K]} z_k = 1$.

A.1.1. SINGLE LEVEL REFORMULATION FOR THE INVERSE MULTI-OBJECTIVE OPTIMIZATION PROBLEM

The parametrized mulobjective optimization problem is

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathbb{R}^n} & \mathbf{f}(\mathbf{x},\theta) \\ s.t. & \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \end{array} \tag{MOP}$$

where

$$\begin{aligned} \mathbf{f}(\mathbf{x},\theta) &= (f_1(\mathbf{x},\theta), f_2(\mathbf{x},\theta),\ldots,f_p(\mathbf{x},\theta))^T \\ \mathbf{g}(\mathbf{x}) &= (g_1(\mathbf{x}),\ldots,g_q(\mathbf{x}))^T \end{aligned}$$

Then, the single level reformulation for the Implicit update in the paper is given in the following

$$
\begin{aligned}
\min_{\mathbf{b}} \quad & \tfrac{1}{2}\|\theta - \theta_t\|_2^2 + \eta_t \sum_{k\in[K]} \|\mathbf{y}_t - \vartheta_k\|_2^2 \\
\text{s.t.} \quad & \theta \in \Theta \\
& \begin{bmatrix} \mathbf{g}(\mathbf{x}_k) \leq \mathbf{0}, \ \ \mathbf{u}_k \geq \mathbf{0} \\ \mathbf{u}_k^T \mathbf{g}(\mathbf{x}_k) = 0 \\ \nabla_{\mathbf{x}_k} w_k^T \mathbf{f}(\mathbf{x}_k, \theta) + \mathbf{u}_k \cdot \nabla_{\mathbf{x}_k} \mathbf{g}(\mathbf{x}_k) = \mathbf{0} \end{bmatrix} & \forall k \in [K] \\
& 0 \leq \vartheta_k \leq M_k z_k & \forall k \in [K] \\
& \mathbf{x}_k - M_k(1 - z_k) \leq \vartheta_k \leq \mathbf{x}_k & \forall k \in [K] \\
& \sum_{k\in[K]} z_k = 1 \\
& \mathbf{x}_k \in \mathbb{R}^n, \ \ \mathbf{u}_k \in \mathbb{R}_+^m, \ \ \mathbf{t}_k \in \{0,1\}^m, \ \ z_k \in \{0,1\} & \forall k \in [K]
\end{aligned}
$$

A.1.2. SINGLE LEVEL REFORMULATION FOR THE INVERSE MULTI-OBJECTIVE QUADRATIC PROBLEM

When the objective functions are quadratic and the feasible region is a polyhedron, the multi-objective optimization has the following form

$$
\begin{aligned}
\min_{\mathbf{x}\in\mathbb{R}^n} \quad & \begin{bmatrix} \frac{1}{2}\mathbf{x}^T Q_1 \mathbf{x} + \mathbf{c}_1^T \mathbf{x} \\ \vdots \\ \frac{1}{2}\mathbf{x}^T Q_p \mathbf{x} + \mathbf{c}_p^T \mathbf{x} \end{bmatrix} \\
s.t. \quad & A\mathbf{x} \geq \mathbf{b}
\end{aligned}
\tag{MQP}
$$

where $Q_l \in \mathbf{S}_+^n$ (the set of symmetric positive semidefinite matrices) for all $l \in [p]$..

When trying to learn $\{\mathbf{c}_l\}_{l\in[p]}$, the single level reformulation for the Implicit update in the paper is given in the following

$$
\begin{aligned}
\min_{\mathbf{c}_l} \quad & \tfrac{1}{2} \sum_{l\in[p]} \|\mathbf{c}_l - \mathbf{c}_l^t\|_2^2 + \eta_t \sum_{k\in[K]} \|\mathbf{y}_t - \vartheta_k\|_2^2 \\
\text{s.t.} \quad & \mathbf{c}_l \in \widetilde{C}_l & \forall l \in [p] \\
& \begin{bmatrix} \mathbf{A}\mathbf{x}_k \geq \mathbf{b}, \ \mathbf{u}_k \geq \mathbf{0} \\ \mathbf{u}_k \leq M\mathbf{t}_k \\ \mathbf{A}\mathbf{x}_k - \mathbf{b} \leq M(1 - \mathbf{t}_k) \\ (w_k^1 Q_1 + \cdots + w_k^p Q_p)\mathbf{x}_i + w_k^1 \mathbf{c}_1 + \cdots + w_k^p \mathbf{c}_p - \mathbf{A}^T \mathbf{u}_k = 0 \end{bmatrix} & \forall k \in [K] \\
& 0 \leq \vartheta_k \leq M_k z_k & \forall k \in [K] \\
& \mathbf{x}_k - M_k(1 - z_k) \leq \vartheta_k \leq \mathbf{x}_k & \forall k \in [K] \\
& \sum_{k\in[K]} z_k = 1 \\
& \mathbf{x}_k \in \mathbb{R}^n, \ \ \mathbf{u}_k \in \mathbb{R}_+^m, \ \ \mathbf{t}_k \in \{0,1\}^m, \ \ z_k \in \{0,1\} & \forall l \in [p] \ \ \forall k \in [K]
\end{aligned}
$$

where $\mathbf{c}_l^t$ is the estimation of $\mathbf{c}_l$ at the tth round, and $\widetilde{C}_l$ is a convex set for each $l \in [p]$.

We have a similar single level reformulation when learning the Right-hand side $\mathbf{b}$. Clearly, this is a Mixed Integer Second Order Cone program(MISOCP) when learning either $\mathbf{c}_l$ or $\mathbf{b}$.

A.2. Omitted Proofs

A.2.1. STRONGLY CONVEX OF $w^T\mathbf{f}(\mathbf{x},\theta)$ AS STATED UNDER ASSUMPTION A.2

Proof By the definition of λ,

$$\begin{aligned}\Big(\nabla w^T\mathbf{f}(\mathbf{y},\theta) - \nabla w^T\mathbf{f}(\mathbf{x},\theta)\Big)^T(\mathbf{y}-\mathbf{x}) &= \Big(\nabla\sum_{l=1}^{p} w_l f_l(\mathbf{y},\theta) - \nabla\sum_{l=1}^{p} w_l f_l(\mathbf{x},\theta_l)\Big)^T(\mathbf{y}-\mathbf{x}) \\ &= \sum_{l=1}^{p} w_l\Big(\nabla f_l(\mathbf{y},\theta_l) - \nabla f_l(\mathbf{x},\theta_l)\Big)^T(\mathbf{y}-\mathbf{x}) \\ &\geq \sum_{l=1}^{p} w_l\lambda_l\|\mathbf{x}-\mathbf{y}\|_2^2 \geq \eta\|\mathbf{x}-\mathbf{y}\|_2^2\sum_{l=1}^{p} w_l \\ &= \lambda\|\mathbf{x}-\mathbf{y}\|_2^2\end{aligned}$$

Thus, $w^T\mathbf{f}(\mathbf{x},\theta)$ is strongly convex for $\mathbf{x}\in\mathbb{R}^n$. ■

A.2.2. PROOF OF LEMMA 6

Proof By Assumption 3.1(b), we know that $S(w,\theta)$ is a single-valued set for each $w\in\mathscr{W}_p$. Thus, $\forall\mathbf{y}\in\mathcal{Y}$, $\forall\theta_1,\theta_2\in\Theta$, $\exists w^1,w^2\in\mathscr{W}_p$, s.t.

$$\mathbf{x}(\theta_1) = S(w^1,\theta_1),\quad \mathbf{x}(\theta_2) = S(w^2,\theta_2)$$

Without of loss of generality, let $l_K(\mathbf{y},\theta_1)\geq l_K(\mathbf{y},\theta_2)$. Then,

$$\begin{aligned}&|l_K(\mathbf{y},\theta_1) - l_K(\mathbf{y},\theta_2)| = l_K(\mathbf{y},\theta_1) - l_K(\mathbf{y},\theta_2) \\ &= \|\mathbf{y}-\mathbf{x}(\theta_1)\|_2^2 - \|\mathbf{y}-\mathbf{x}(\theta_2)\|_2^2 \\ &= \|\mathbf{y}-S(w^1,\theta_1)\|_2^2 - \|\mathbf{y}-S(w^2,\theta_2)\|_2^2 \\ &\leq \|\mathbf{y}-S(w^2,\theta_1)\|_2^2 - \|\mathbf{y}-S(w^2,\theta_2)\|_2^2 \\ &= \langle S(w^2,\theta_2)-S(w^2,\theta_1), 2\mathbf{y}-S(w^2,\theta_1)-S(w^2,\theta_2)\rangle \\ &\leq 2(B+R)\|S(w^2,\theta_2)-S(w^2,\theta_1)\|_2\end{aligned} \tag{5}$$

The last inequality is due to Cauchy-Schwartz inequality and the Assumptions 3.1(a), that is

$$\|2\mathbf{y}-S(w^2,\theta_1)-S(w^2,\theta_2)\|_2 \leq 2(B+R) \tag{6}$$

Next, we will apply Proposition 6.1 in [5] to bound $\|S(w^2,\theta_2)-S(w^2,\theta_1)\|_2$.
Under Assumptions 3.1 - 3.2, the conditions of Proposition 6.1 in [5] are satisfied. Therefore,

$$\|S(w^2,\theta_2)-S(w^2,\theta_1)\|_2 \leq \frac{2\kappa}{\lambda}\|\theta_1-\theta_2\|_2 \tag{7}$$

Plugging equation 6 and equation 7 in equation 5 yields the claim. ■

A.3. Assumptions

In the following, we make a few assumptions to simplify our understanding, which are actually mild and appear often in the literature.

Assumption A.1 *Set Θ is a convex compact set. There exists $D > 0$ such that $\|\theta\|_2 \leq D$ for all $\theta \in \Theta$. In addition, for each $\theta \in \Theta$, both $\mathbf{f}(\mathbf{x}, \theta)$ and $\mathbf{g}(\mathbf{x}, \theta)$ are convex in $\mathbf{x}$.*

Assumption A.2

(a) *$X(\theta)$ is closed, and has a nonempty relative interior. $X(\theta)$ is also bounded. Namely, there exists $B > 0$ such that $\|\mathbf{x}\|_2 \leq B$ for all $\mathbf{x} \in X(\theta)$. The support $\mathcal{Y}$ of the noisy decisions $\mathbf{y}$ is contained within a ball of radius R almost surely, where $R < \infty$. In other words, $\mathbb{P}(\|\mathbf{y}\|_2 \leq R) = 1$.*

(b) *Each function in $\mathbf{f}$ is strongly convex on $\mathbb{R}^n$, that is for each $l \in [p]$, $\exists \lambda_l > 0$, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$*

$$\left(\nabla f_l(\mathbf{y}, \theta_l) - \nabla f_l(\mathbf{x}, \theta_l) \right)^T (\mathbf{y} - \mathbf{x}) \geq \lambda_l \|\mathbf{x} - \mathbf{y}\|_2^2.$$

Regarding Assumption A.2.(a), assuming that the feasible region is closed and bounded is very common in inverse optimization. The finite support of the observations is needed since we do not hope outliers have too many impacts in our learning. Let $\lambda = \min_{l \in [p]} \{\lambda_l\}$. It follows that $w^T \mathbf{f}(\mathbf{x}, \theta)$ is strongly convex with parameter λ for $w \in \mathscr{W}_p$. Therefore, Assumption A.2.(b) ensures that $S(w, \theta)$ is a single-valued set for each w.

The performance of the algorithm also depends on how the change of θ affects the objective values. For $\forall w \in \mathscr{W}_p, \theta_1 \in \Theta, \theta_2 \in \Theta$, we consider the following function

$$h(\mathbf{x}, w, \theta_1, \theta_2) = w^T \mathbf{f}(\mathbf{x}, \theta_1) - w^T \mathbf{f}(\mathbf{x}, \theta_2).$$

Assumption A.3 *$\exists \kappa > 0$, $\forall w \in \mathscr{W}_p$, $h(\cdot, w, \theta_1, \theta_2)$ is κ-Lipschitz continuous on $\mathcal{Y}$. That is,*

$$|h(\mathbf{x}, w, \theta_1, \theta_2) - h(\mathbf{y}, w, \theta_1, \theta_2)| \leq \kappa \|\theta_1 - \theta_2\|_2 \|\mathbf{x} - \mathbf{y}\|_2, \forall \mathbf{x}, \mathbf{y} \in \mathcal{Y}.$$

Basically, this assumption says that the objective functions will not change much when either the parameter θ or the variable $\mathbf{x}$ is perturbed. It actually holds in many common situations, including the multi-objective linear program and multi-objective quadratic program.

From now on, given any $\mathbf{y} \in \mathcal{Y}, \theta \in \Theta$, we denote $\mathbf{x}(\theta)$ the efficient point in $X_E(\theta)$ that is closest to $\mathbf{y}$. Namely, $l(\mathbf{y}, \theta) = \|\mathbf{y} - \mathbf{x}(\theta)\|_2^2$.

Lemma 6 *Under Assumptions A.2 - A.3, the loss function $l(\mathbf{y}, \theta)$ is uniformly $\frac{4(B+R)\kappa}{\lambda}$-Lipschitz continuous in θ. That is, $\forall \mathbf{y} \in \mathcal{Y}, \forall \theta_1, \theta_2 \in \Theta$, we have*

$$|l(\mathbf{y}, \theta_1) - l(\mathbf{y}, \theta_2)| \leq \frac{4(B+R)\kappa}{\lambda} \|\theta_1 - \theta_2\|_2.$$

The key point in proving Lemma 6 is the observation that the perturbation of $S(w, \theta)$ due to θ is bounded by the perturbation of θ by applying Proposition 6.1 in Bonnans and Shapiro [5]. Details of the proof are given in Appendix.

Assumption A.4 *For MOP,* $\forall \mathbf{y} \in \mathcal{Y}, \forall \theta_1, \theta_2 \in \Theta$, $\forall \alpha, \beta \geq 0$ *s.t.* $\alpha + \beta = 1$, *we have either of the following:*

(a) *if* $\mathbf{x}_1 \in X_E(\theta_1)$*, and* $\mathbf{x}_2 \in X_E(\theta_2)$*, then* $\alpha\mathbf{x}_1 + \beta\mathbf{x}_2 \in X_E(\alpha\theta_1 + \beta\theta_2)$.

(b) $\|\alpha\mathbf{x}(\theta_1) + \beta\mathbf{x}(\theta_2) - \mathbf{x}(\alpha\theta_1 + \beta\theta_2)\|_2 \leq \alpha\beta\|\mathbf{x}(\theta_1) - \mathbf{x}(\theta_2)\|_2/(2(B+R))$.

The definition of $\mathbf{x}(\theta_1), \mathbf{x}(\theta_2)$ and $\mathbf{x}(\alpha\theta_1 + \beta\theta_2)$ is given before Lemma 6. This assumption requires the convex combination of $\mathbf{x}_1 \in X_E(\theta_1)$, and $\mathbf{x}_2 \in X_E(\theta_2)$ belongs to $X_E(\alpha\theta_1 + \beta\theta_2)$. Or there exists an efficient point in $X_E(\alpha\theta_1 + \beta\theta_2)$ close to the convex combination of $\mathbf{x}(\theta_1)$ and $\mathbf{x}(\theta_2)$. Examples are given in Appendix.

A.3.1. PROOF OF THEOREM 5

Proof We will extend Theorem 3.2 in [16] to prove our theorem.

Let $G_t(\theta) = \frac{1}{2}\|\theta - \theta_t\|_2^2 + \eta_t l(\mathbf{y}_t, \theta)$.

We will now show the loss function is convex. The first step is to show that if Assumption 3.3 holds, then the loss function $l(\mathbf{y}, \theta)$ is convex in θ.

First, suppose Assumption 3.3(a) hold. Then,

$$\begin{aligned}
& \alpha l(\mathbf{y}, \theta_1) + \beta l(\mathbf{y}, \theta_2) - l(\mathbf{y}, \alpha\theta_1 + \beta\theta_2) \\
= \; & \alpha\|\mathbf{y} - \mathbf{x}(\theta_1)\|_2^2 + \beta\|\mathbf{y} - \mathbf{x}(\theta_2)\|_2^2 - \|\mathbf{y} - \mathbf{x}(\alpha\theta_1 + \beta\theta_2)\|_2^2 \\
\geq \; & \alpha\|\mathbf{y} - \mathbf{x}(\theta_1)\|_2^2 + \beta\|\mathbf{y} - \mathbf{x}(\theta_2)\|_2^2 - \|\mathbf{y} - \alpha\mathbf{x}(\theta_1) - \beta\mathbf{x}(\theta_2)\|_2^2 \quad \text{(By Assumption 3.3(a))} \\
= \; & \alpha\beta\|\mathbf{x}(\theta_1) - \mathbf{x}(\theta_2)\|_2^2 \\
\geq \; & 0
\end{aligned} \tag{8}$$

Second, suppose Assumption 3.3(b) holds. Then,

$$\begin{aligned}
& \alpha l(\mathbf{y}, \theta_1) + \beta l(\mathbf{y}, \theta_2) - l(\mathbf{y}, \alpha\theta_1 + \beta\theta_2) \\
= \; & \alpha\|\mathbf{y} - \mathbf{x}(\theta_1)\|_2^2 + \beta\|\mathbf{y} - \mathbf{x}(\theta_2)\|_2^2 - \|\mathbf{y} - \mathbf{x}(\alpha\theta_1 + \beta\theta_2)\|_2^2 \\
= \; & \alpha\|\mathbf{y} - \mathbf{x}(\theta_1)\|_2^2 + \beta\|\mathbf{y} - \mathbf{x}(\theta_2)\|_2^2 - \|\mathbf{y} - \alpha\mathbf{x}(\theta_1) - \beta\mathbf{x}(\theta_2)\|_2^2 \\
& + \|\mathbf{y} - \alpha\mathbf{x}(\theta_1) - \beta\mathbf{x}(\theta_2)\|_2^2 - \|\mathbf{y} - \mathbf{x}(\alpha\theta_1 + \beta\theta_2)\|_2^2 \\
= \; & \alpha\beta\|\mathbf{x}(\theta_1) - \mathbf{x}(\theta_2)\|_2^2 + \|\mathbf{y} - \alpha\mathbf{x}(\theta_1) - \beta\mathbf{x}(\theta_2)\|_2^2 - \|\mathbf{y} - \mathbf{x}(\alpha\theta_1 + \beta\theta_2)\|_2^2 \\
= \; & \alpha\beta\|\mathbf{x}(\theta_1) - \mathbf{x}(\theta_2)\|_2^2 - \langle \alpha\mathbf{x}(\theta_1) + \beta\mathbf{x}(\theta_2) - \mathbf{x}(\alpha\theta_1 + \beta\theta_2), 2\mathbf{y} - \mathbf{x}(\alpha\theta_1 + \beta\theta_2) - \alpha\mathbf{x}(\theta_1) - \beta\mathbf{x}(\theta_2)\rangle \\
\geq \; & \alpha\beta\|\mathbf{x}(\theta_1) - \mathbf{x}(\theta_2)\|_2^2 - \|\alpha\mathbf{x}(\theta_1) + \beta\mathbf{x}(\theta_2) - \mathbf{x}(\alpha\theta_1 + \beta\theta_2)\|_2 \|2\mathbf{y} - \mathbf{x}(\alpha\theta_1 + \beta\theta_2) - \alpha\mathbf{x}(\theta_1) - \beta\mathbf{x}(\theta_2)\|_2
\end{aligned} \tag{9}$$

The last inequality is by Cauchy-Schwartz inequality. Note that

$$\begin{aligned}
& \|\alpha\mathbf{x}(\theta_1) + \beta\mathbf{x}(\theta_2) - \mathbf{x}(\alpha\theta_1 + \beta\theta_2)\|_2 \|2\mathbf{y} - \mathbf{x}(\alpha\theta_1 + \beta\theta_2) - \alpha\mathbf{x}(\theta_1) - \beta\mathbf{x}(\theta_2)\|_2 \\
\leq \; & 2(B+R)\|\alpha\mathbf{x}(\theta_1) + \beta\mathbf{x}(\theta_2) - \mathbf{x}(\alpha\theta_1 + \beta\theta_2)\|_2 \\
\leq \; & \alpha\beta\|\mathbf{x}(\theta_1) - \mathbf{x}(\theta_2)\|_2 \qquad \text{(By Assumption 3.3(b))}
\end{aligned} \tag{10}$$

Plugging equation 10 in equation 9 yields the result.

Using Theorem 3.2. in [16], for $\alpha_t \leq \frac{G_t(\theta_{t+1})}{G_t(\theta_t)}$, we have

$$R_T \leq \begin{array}{l} \sum_{t=1}^{T} \frac{1}{\eta_t}(1-\alpha_t)\eta_t l(\mathbf{y}_t, \theta_t) \\ +\frac{1}{2\eta_t}(\|\theta_t - \theta^*\|_2^2 - \|\theta_{t+1} - \theta^*\|_2^2) \end{array} \tag{11}$$

Notice that

$$\begin{array}{rl} & G_t(\theta_t) - G_t(\theta_{t+1}) \\ = & \eta_t(l(\mathbf{y}_t, \theta_t) - l(\mathbf{y}_t, \theta_{t+1})) - \frac{1}{2}\|\theta_t - \theta_{t+1}\|_2^2 \\ \leq & \frac{4(B+R)\kappa\eta_t}{\lambda}\|\theta_t - \theta_{t+1}\|_2 - \frac{1}{2}\|\theta_t - \theta_{t+1}\|_2^2 \\ \leq & \frac{8(B+R)^2\kappa^2\eta_t^2}{\lambda^2} \end{array} \tag{12}$$

The first inequality follows by applying Lemma 3.1.
Let $\alpha_t = \frac{R_t(\theta_{t+1})}{R_t(\theta_t)}$. Using equation 12, we have

$$\begin{array}{rl} (1-\alpha_t)\eta_t l(\mathbf{y}_t, \theta_t) & = (1-\alpha_t)G_t(\theta_t) \\ & = G_t(\theta_t) - G_t(\theta_{t+1}) \\ & \leq \frac{8(B+R)^2\kappa^2\eta_t^2}{\lambda^2} \end{array} \tag{13}$$

Plug equation 13 in equation 11, and note the telescoping sum,

$$\begin{aligned} R_T \leq & \sum_{t=1}^{T} \frac{8(B+R)^2\kappa^2\eta_t}{\lambda^2} \\ & + \sum_{t=1}^{T} \frac{1}{2\eta_t}(\|\theta_t - \theta^*\|_2^2 - \|\theta_{t+1} - \theta^*\|_2^2) \end{aligned}$$

Setting $\eta_t = \frac{D\lambda}{2(B+R)\kappa\sqrt{2t}}$, we can simplify the second summation to $\frac{D(B+R)\kappa\sqrt{2}}{\lambda}$ since the sum telescopes and $\theta_1 = \mathbf{0}$, $\|\theta^*\|_2 \leq D$. The first sum simplifies using $\sum_{t=1}^{T} \frac{1}{\sqrt{t}} \leq 2\sqrt{T} - 1$ to obtain the result

$$R_T \leq \frac{4\sqrt{2}(B+R)D\kappa}{\lambda}\sqrt{T}.$$

■

A.4. Omitted Examples

A.4.1. EXAMPLES FOR WHICH ASSUMPTION A.4 HOLDS

Consider for example the following quadratic program

$$\begin{array}{cl} \min\limits_{\mathbf{x}\in\mathbb{R}^n} & \begin{pmatrix} \mathbf{x}^T\mathbf{x} - 2\theta_1^T\mathbf{x} \\ \mathbf{x}^T\mathbf{x} - 2\theta_2^T\mathbf{x} \end{pmatrix} \\ s.t. & 0 \leq \mathbf{x} \leq 10 \end{array}$$

One can check that Assumption 3.3 (a) is indeed satisfied. For example, let $n = 1$. Then, W.L.O.G, let $\theta_1 \leq \theta_2$. Then, $X_E(\theta) = [\theta_1, \theta_2]$. Consider two parameters that $\theta^1 = (\theta_1^1, \theta_2^1), \theta^2 = (\theta_1^2, \theta_2^2) \in [0, 10]^2$. For all $\alpha \in [0, 1]$,

$$X_E(\alpha\theta^1 + (1-\alpha)\theta^2) = [\alpha\theta_1^1 + (1-\alpha)\theta_1^2, \alpha\theta_2^1 + (1-\alpha)\theta_2^2]$$

Although tedious, one can check that one can check that Assumption 3.3 (a) is indeed satisfied.

A.5. Data for the Portfolio optimization problem

Table 1: True Expected Return

Security	1	2	3	4	5	6	7	8
Expected Return	0.1791	0.1143	0.1357	0.0837	0.1653	0.1808	0.0352	0.0368

Table 2: True Return Covariances Matrix

Security	1	2	3	4	5	6	7	8
1	0.1641	0.0299	0.0478	0.0491	0.058	0.0871	0.0603	0.0492
2	0.0299	0.0720	0.0511	0.0287	0.0527	0.0297	0.0291	0.0326
3	0.0478	0.0511	0.0794	0.0498	0.0664	0.0479	0.0395	0.0523
4	0.0491	0.0287	0.0498	0.1148	0.0336	0.0503	0.0326	0.0447
5	0.0580	0.0527	0.0664	0.0336	0.1073	0.0483	0.0402	0.0533
6	0.0871	0.0297	0.0479	0.0503	0.0483	0.1134	0.0591	0.0387
7	0.0603	0.0291	0.0395	0.0326	0.0402	0.0591	0.0704	0.0244
8	0.0492	0.0326	0.0523	0.0447	0.0533	0.0387	0.0244	0.1028

A.6. Approximation error

Theorem 7 *Under Assumption A.2, we have that* $\forall \mathbf{y} \in \mathcal{Y}, \forall \theta \in \Theta$,

$$0 \leq l_K(\mathbf{y}, \theta) - l(\mathbf{y}, \theta) \leq \frac{4(B+R)\zeta}{\lambda} \cdot \frac{\sqrt{2p}}{\Lambda - 1},$$

where

$$K = \frac{(\Lambda + p - 2)!}{(\Lambda - 1)!(p-1)!}, \zeta = \max_{l \in [p], \mathbf{x} \in X(\theta), \theta \in \Theta} |f_l(\mathbf{x}, \theta)|.$$

Furthermore,

$$0 \leq l_K(\mathbf{y}, \theta) - l(\mathbf{y}, \theta) \leq \frac{16e(B+R)\zeta}{\lambda} \cdot \frac{1}{K^{\frac{1}{p-1}}}.$$

Thus, the surrogate loss function uniformly converges to the loss function at the rate of $\mathcal{O}(1/K^{\frac{1}{p-1}})$. Note that this rate exhibits a dependence on the number of objective functions p. As p increases, we might require (approximately) exponentially more weight samples $\{w_K\}_{k \in [K]}$ to achieve an approximation accuracy. In fact, this phenomenon is a reflection of *curse of dimensionality* [13], a principle that estimation becomes exponentially harder as the number of dimension increases. In particular, the dimension here is the number of objective functions p. Naturally, one way to deal with the curse of dimensionality is to employ dimension reduction techniques in statistics to find a low-dimensional representation of the objective functions.

Example 1 *When $p = 2$, MOP is a bi-objective decision making problem. Then, Theorem 7 shows that $l_K(\mathbf{y}, \theta) - l(\mathbf{y}, \theta)$ is of $\mathcal{O}(1/K)$. That is, $l_K(\mathbf{y}, \theta)$ asymptotically converges to $l(\mathbf{y}, \theta)$ sublinearly.*

Proof By definition,

$$l_K(\mathbf{y}, \theta) - l(\mathbf{y}, \theta) = \min_{\mathbf{x} \in \bigcup\limits_{k \in [K]} S(w_k, \theta)} \|\mathbf{y} - \mathbf{x}\|_2^2 - \min_{\mathbf{x} \in X_E(\theta)} \|\mathbf{y} - \mathbf{x}\|_2^2 \geq 0.$$

Let $\|\mathbf{y} - S(w_k^{\mathbf{y}}, \theta)\|_2^2 = \min_{\mathbf{x} \in \bigcup\limits_{k \in [K]} S(w_k, \theta)} \|\mathbf{y} - \mathbf{x}\|_2^2$, and $\|\mathbf{y} - S(w^{\mathbf{y}}, \theta)\|_2^2 = \min_{\mathbf{x} \in X_E(\theta)} \|\mathbf{y} - \mathbf{x}\|_2^2$. Let $w_{k'}^{\mathbf{y}}$ be the closest weight sample among $\{w_k\}_{k \in [K]}$ to $w^{\mathbf{y}}$. Then,

$$\begin{aligned}
l_K(\mathbf{y}, \theta) - l(\mathbf{y}, \theta) &= \|\mathbf{y} - S(w_k^{\mathbf{y}}, \theta)\|_2^2 - \|\mathbf{y} - S(w^{\mathbf{y}}, \theta)\|_2^2 \\
&\leq \|\mathbf{y} - S(w_{k'}^{\mathbf{y}}, \theta)\|_2^2 - \|\mathbf{y} - S(w^{\mathbf{y}}, \theta)\|_2^2 \\
&= \left(2\mathbf{y} - S(w_{k'}^{\mathbf{y}}, \theta) - S(w^{\mathbf{y}}, \theta)\right)^T \left(S(w^{\mathbf{y}}, \theta) - S(w_{k'}^{\mathbf{y}}, \theta)\right) \\
&\leq \|2\mathbf{y} - S(w_{k'}^{\mathbf{y}}, \theta) - S(w^{\mathbf{y}}, \theta)\|_2 \|S(w^{\mathbf{y}}, \theta) - S(w_{k'}^{\mathbf{y}}, \theta)\|_2 \\
&\leq 2(B + R)\|S(w^{\mathbf{y}}, \theta) - S(w_{k'}^{\mathbf{y}}, \theta)\|_2 \\
&\leq \tfrac{4(B+R)\zeta\sqrt{p}}{\lambda} \cdot \|w^{\mathbf{y}} - w_{k'}^{\mathbf{y}}\|_2,
\end{aligned} \tag{14}$$

where $\zeta = \max_{l \in [p], \mathbf{x} \in X(\theta), \theta \in \Theta} |f_l(\mathbf{x}, \theta)|$. The third inequality is due to Cauchy Schwarz inequality. Under Assumption A.2, we can apply Lemma 4 in [7] to yield the last inequality.

Next, we will show that $\forall w \in \mathscr{W}_p$, the distance between w and its closest weight sample among $\{w_k\}_{k \in [K]}$ is upper bounded by the function of K and p and nothing else. More precisely, we will show that

$$\sup_{w \in \mathscr{W}_p} \min_{k \in [K]} \|w - w_k\|_2 \leq \frac{\sqrt{2}}{\Lambda - 1}. \tag{15}$$

Here, Λ is the number of evenly spaced weight samples between any two extreme points of $\mathscr{W}_p$.

Note that $\{w_k\}_{k \in [K]}$ are evenly sampled from $\mathscr{W}_p$, and that the distance between any two extreme points of $\mathscr{W}_p$ equals to $\sqrt{2}$. Hence, the distances between any two neighboring weight samples are equal and can be calculated as the distance between any two extreme points of $\mathscr{W}_p$ divided by $\Lambda - 1$. Proof of equation 15 can be done by further noticing that the distance between any w and $\{w_k\}_{k \in [K]}$ is upper bounded by the distances between any two neighboring weight samples.

Combining equation 14 and equation 15 yields that

$$0 \leq l_K(\mathbf{y}, \theta) - l(\mathbf{y}, \theta) \leq \frac{4(B + R)\zeta}{\lambda} \cdot \frac{\sqrt{2p}}{\Lambda - 1}, \tag{16}$$

Then, we can prove that the total number of weight samples K and Λ has the following relationship:

$$K = \binom{\Lambda + p - 2}{p - 1} \tag{17}$$

Proof of equation 17 can be done by induction with respect to p. Obviously, equation 17 holds when $p = 2$ as $K = \Lambda$. Assume equation 17 holds for the $\leq p - 1$ cases. For ease of notation, denote

$$K_p^\Lambda = \binom{\Lambda + p - 2}{p - 1}.$$

Then, for the p case, we note that the weight samples can be classified into two categories: $w_p = 0$; $w_p > 0$. For $w_p = 0$, the number of weight samples is simply K_{p-1}^Λ. For $w_p > 0$, the number of weight samples is $K_p^{\Lambda-1}$. Thus,

$$K = K_{p-1}^\Lambda + K_p^{\Lambda-1}. \tag{18}$$

Iteratively expanding $K_p^{\Lambda-1}$ through the same argument as equation 17 and using the fact that

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k},$$

we have

$$\begin{aligned} K \quad &= K_{p-1}^\Lambda + K_p^{\Lambda-1} = K_{p-1}^\Lambda + K_{p-1}^{\Lambda-1} + K_p^{\Lambda-2} \\ &\vdots \\ &= K_{p-1}^\Lambda + K_{p-1}^{\Lambda-1} + \cdots + K_{p-1}^2 + K_p^1 \\ &= \binom{\Lambda + p - 3}{p - 2} + \binom{\Lambda + p - 4}{p - 2} + \cdots + \binom{p-1}{p-2} + \binom{p-1}{p-1} \\ &= \tfrac{(\Lambda+p-2)!}{(\Lambda-1)!(p-1)!} \end{aligned} \tag{19}$$

To this end, we complete the proof of equation 17.

Furthermore, we notice that

$$K = \frac{(\Lambda + p - 2)!}{(\Lambda - 1)!(p - 1)!} \leq \frac{(\Lambda + p - 2)^{p-1}}{(p - 1)!} < \left(\frac{\Lambda + p - 2}{p - 1}\right)^{p-1} \cdot e^{p-1}.$$

Then, when $\Lambda \geq p (K \geq 2^{p-1})$, through simple algebraic calculation we have

$$\frac{e}{K^{\frac{1}{p-1}}} > \frac{p - 1}{\Lambda + p - 2} > \frac{1}{4} \cdot \frac{p}{\Lambda - 1} \tag{20}$$

We complete the proof by combining equation 16 and equation 20 and noticing that $\sqrt{2p} \leq p$. ■

A.7. Real-world case: learning expected returns in portfolio optimization

We next consider noisy decisions arising from different investors in a stock market. More precisely, we consider a portfolio selection problem, where investors need to determine the fraction of their wealth to invest in each security to maximize the total return and minimize the total risk. The process typically involves the cooperation between an investor and a portfolio analyst, where the analyst provides an efficient frontier on a certain set of securities to the investor and then the investor selects a portfolio according to her preference to the returns and risks. The classical Markovitz mean-variance portfolio selection [17] in the following is used by analysts.

$$\begin{array}{lll} \min & \begin{pmatrix} f_1(\mathbf{x}) & = -\mathbf{r}^T\mathbf{x} \\ f_2(\mathbf{x}) & = \mathbf{x}^T Q \mathbf{x} \end{pmatrix} & \\ s.t. & 0 \leq x_i \leq b_i, & \forall i \in [n], \\ & \sum_{i=1}^{n} x_i = 1, & \end{array}$$

where $\mathbf{r} \in \mathbb{R}^n_+$ is a vector of individual security expected returns, $Q \in \mathbb{R}^{n \times n}$ is the covariance matrix of securities returns, $\mathbf{x}$ is a portfolio specifying the proportions of capital to be invested in the different securities, and b_i is an upper bound on the proportion of security i, $\forall i \in [n]$.

Dataset: The dataset is derived from monthly total returns of 30 stocks from a blue-chip index which tracks the performance of top 30 stocks in the market when the total investment universe consists of thousands of assets. The true expected returns and true return covariance matrix for the first 8 securities are given in the Appendix.

Details for generating the portfolios are provided in Appendix. The portfolios on the efficient frontier are plot in Figure **??**. The learning rate is set to $\eta_t = 5/\sqrt{t}$. At each round t, we solve 4 using parallel computing. In Table 3 we list the estimation error and estimated expected returns for different K. The estimation error becomes smaller when K increases, indicating that we have a better approximation accuracy of the efficient set when using a larger K. We also plot the estimated efficient frontier using the estimated $\hat{\mathbf{r}}$ for $K = 41$ in Figure 2 (a). We can see that the estimated efficient frontier is very close to the real one, showing that our algorithm works quite well in learning expected returns in portfolio optimization. We also plot our estimation on the distribution of the weight of $f_1(\mathbf{x})$ among the 1000 decision makers. As shown in Figure 2 (b), the distribution follows roughly normal distribution. We apply Chi-square goodness-of-fit tests to support our hypotheses.

Table 3: Estimation Error for Different K

K	6	11	21	41
$\|\hat{\mathbf{r}} - \mathbf{r}_{true}\|_2$	0.1270	0.1270	0.0420	0.0091

Figure 2: Learning the expected return of a Portfolio optimization problem over $T = 1000$ rounds with $K = 41$. (a) The red line indicates the real efficient frontier. The blue dots indicate the estimated efficient frontier using the estimated expected return for $K = 41$. (b) Each bar represents the proportion of the 1000 decision makers that has the corresponding weight for $f_1(\mathbf{x})$.