Inverse Optimization with Online Data and Multiobjectives: Models, Insights and Algorithms

Bo Zeng

Department of Industrial Engineering, University of Pittsburgh

Joint work with Chaosheng Dong, currently at Amazon

November 13, 2020, Texas A&M Institute of Data Science

University of Pittsburgh

Introduction

- We (humans, enterprises and organizations) are decision makers, making various decisions every moment everywhere.

Introduction

- We (humans, enterprises and organizations) are decision makers, making various decisions every moment everywhere.
- Decision makers are driven by their interests, desires, preferences, or utility in general, subject to different restrictions

Introduction

- We (humans, enterprises and organizations) are decision makers, making various decisions every moment everywhere.
- Decision makers are driven by their interests, desires, preferences, or utility in general, subject to different restrictions
- Decisions, represented in the form of choices, behaviors, operations et al., are generally observable
  $\Rightarrow$ stored as data.

Introduction

- We (humans, enterprises and organizations) are decision makers, making various decisions every moment everywhere.
- Decision makers are driven by their interests, desires, preferences, or utility in general, subject to different restrictions
- Decisions, represented in the form of choices, behaviors, operations et al., are generally observable
  $\Rightarrow$ stored as data.
- For a service provider/manufacturer/supplier, developing a sound understanding on decision makers (i.e., their interests, desires, preferences, and/or restrictions) is critical and fundamental,
  $\Rightarrow$ how to convert data into information or knowledge?

Introduction

- We (humans, enterprises and organizations) are decision makers, making various decisions every moment everywhere.
- Decision makers are driven by their interests, desires, preferences, or utility in general, subject to different restrictions
- Decisions, represented in the form of choices, behaviors, operations et al., are generally observable
  $\Rightarrow$ stored as data.
- For a service provider/manufacturer/supplier, developing a sound understanding on decision makers (i.e., their interests, desires, preferences, and/or restrictions) is critical and fundamental,
  $\Rightarrow$ how to convert data into information or knowledge?
- Example: you notice from your office that people are using umbrella. Using umbrella indicates people protect themselves from rain. So, an inference is that it is raining now and people do not like wet clothes.

Introduction

- We believe that decision makers are rational, i.e., they acquire and carry out optimal decisions in their decision making problems.

Introduction

- We believe that decision makers are rational, i.e., they acquire and carry out optimal decisions in their decision making problems.
- Decision makers are concerned with

Introduction

- We believe that decision makers are rational, i.e., they acquire and carry out optimal decisions in their decision making problems.
- Decision makers are concerned with
  - a single objective, e.g., the shortest distance

Introduction

- We believe that decision makers are rational, i.e., they acquire and carry out optimal decisions in their decision making problems.
- Decision makers are concerned with
  - a single objective, e.g., the shortest distance
  - multiple objectives, e.g., risk and returns

Introduction

- We believe that decision makers are rational, i.e., they acquire and carry out optimal decisions in their decision making problems.
- Decision makers are concerned with
  - a single objective, e.g., the shortest distance
  - multiple objectives, e.g., risk and returns
- To understand those decision makers, the fundamental issue:
  how to recover the decision making problem (DMP) from observed decisions, e.g., utility functions, restrictions and the overall decision making scheme.

Introduction

- We believe that decision makers are rational, i.e., they acquire and carry out optimal decisions in their decision making problems.
- Decision makers are concerned with
  - a single objective, e.g., the shortest distance
  - multiple objectives, e.g., risk and returns
- To understand those decision makers, the fundamental issue: how to recover the decision making problem (DMP) from observed decisions, e.g., utility functions, restrictions and the overall decision making scheme.
- Inverse Optimization – a data-driven learning approach from observed decisions.

What is inverse optimization problem (IOP)?

- Given a set of observations that are (probably noisy or suboptimal) optimal solutions collected from the decision maker under different external signals, the inverse optimization model is to infer the parameter θ of the DMP with a single objective.

[1] **mas1995microeconomic**.

What is inverse optimization problem (IOP)?

- Given a set of observations that are (probably noisy or suboptimal) optimal solutions collected from the decision maker under different external signals, the inverse optimization model is to infer the parameter θ of the DMP with a single objective.
- Consider the consumer's behavior problem in a market with n products. The prices for the products are denoted by $\mathbf{p}_t$ which varies over different time $t \in [T]$. The consumer's decision making problem can be stated as the following utility maximization problem[1]

$$\begin{aligned} \max_{\mathbf{x} \in \mathbb{R}^n_+} \quad & u(\mathbf{x}) \\ s.t. \quad & \mathbf{p}_t^T \mathbf{x} \le b \end{aligned} \qquad \text{UMP}$$

where $\mathbf{p}_t^T \mathbf{x} \le b$ is the budget constraint at time t.

[1] **mas1995microeconomic**.

Motivation of our research

- Tools of traditional IOP theory (typically for batch setting) have not proven fully applicable to support recent attempts in AI to automate the elicitation of human decision maker's preferences.

[2] **aswani2016inverse**.

Motivation of our research

- Tools of traditional IOP theory (typically for batch setting) have not proven fully applicable to support recent attempts in AI to automate the elicitation of human decision maker's preferences.

[2] **aswani2016inverse**.

Motivation of our research

- Tools of traditional IOP theory (typically for batch setting) have not proven fully applicable to support recent attempts in AI to automate the elicitation of human decision maker's preferences.

[2] **aswani2016inverse**.

Motivation of our research

- Tools of traditional IOP theory (typically for batch setting) have not proven fully applicable to support recent attempts in AI to automate the elicitation of human decision maker's preferences.
  - Recommender systems utilized by online retailers to increase product sales: they elicit a user's preferences or restrictions from a sequence of historical records of her purchasing behaviors, and then make predictions about future shopping decisions.

[2] **aswani2016inverse**.

Motivation of our research

- Tools of traditional IOP theory (typically for batch setting) have not proven fully applicable to support recent attempts in AI to automate the elicitation of human decision maker's preferences.
  - Recommender systems utilized by online retailers to increase product sales: they elicit a user's preferences or restrictions from a sequence of historical records of her purchasing behaviors, and then make predictions about future shopping decisions.
  - Access to large data sets (online/sequential data).

[2] **aswani2016inverse**.

Motivation of our research

- Tools of traditional IOP theory (typically for batch setting) have not proven fully applicable to support recent attempts in AI to automate the elicitation of human decision maker's preferences.
  - Recommender systems utilized by online retailers to increase product sales: they elicit a user's preferences or restrictions from a sequence of historical records of her purchasing behaviors, and then make predictions about future shopping decisions.
  - Access to large data sets (online/sequential data).

[2] **aswani2016inverse**.

Motivation of our research

- Tools of traditional IOP theory (typically for batch setting) have not proven fully applicable to support recent attempts in AI to automate the elicitation of human decision maker's preferences.
  - Recommender systems utilized by online retailers to increase product sales: they elicit a user's preferences or restrictions from a sequence of historical records of her purchasing behaviors, and then make predictions about future shopping decisions.
  - Access to large data sets (online/sequential data).

- However, using traditional IOP to extract users' preferences or restrictions is time consuming, since it is NP-hard (computationally intractable)[2].

---

[2] **aswani2016inverse**.

Motivation of our research

- To fully unlock the potential of inverse optimization, elicit decision maker's preferences or restrictions through online learning.

Motivation of our research

- To fully unlock the potential of inverse optimization, elicit decision maker's preferences or restrictions through online learning.
- We formulate an IOP considering noisy data, develop an online learning algorithm to derive unknown parameters in objective function and/or constraints.

Motivation of our research

- To fully unlock the potential of inverse optimization, elicit decision maker's preferences or restrictions through online learning.
- We formulate an IOP considering noisy data, develop an online learning algorithm to derive unknown parameters in objective function and/or constraints.
- One key feature: it should incorporates sequentially arrived observations into this model, without keeping them in memory, to realize incremental elicitation, revision and reuse of old inferences.

Decision making problem

- We consider a family of parameterized optimization problem, in which $\mathbf{x} \in \mathbb{R}^n$ is the decision variable, $u \in \mathbb{R}^m$ is the external signal, and $\theta \in \Theta$ is the parameter.

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} \quad & f(\mathbf{x}, u, \theta) \\ s.t. \quad & \mathbf{g}(\mathbf{x}, u, \theta) \leq \mathbf{0} \end{aligned} \qquad \text{DMP}$$

where $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \mapsto \mathbb{R}$ is a real-valued function, and $\mathbf{g} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \mapsto \mathbb{R}^q$ is a vector-valued function.

Decision making problem

- We consider a family of parameterized optimization problem, in which $\mathbf{x} \in \mathbb{R}^n$ is the decision variable, $u \in \mathbb{R}^m$ is the external signal, and $\theta \in \Theta$ is the parameter.

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} \quad & f(\mathbf{x}, u, \theta) \\ s.t. \quad & \mathbf{g}(\mathbf{x}, u, \theta) \leq \mathbf{0} \end{aligned} \qquad \text{DMP}$$

  where $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \mapsto \mathbb{R}$ is a real-valued function, and $\mathbf{g} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \mapsto \mathbb{R}^q$ is a vector-valued function.
- $X(u, \theta) = \{x \in \mathbb{R}^n : \mathbf{g}(\mathbf{x}, u, \theta) \leq \mathbf{0}\}$, the feasible region of DMP.

Decision making problem

- We consider a family of parameterized optimization problem, in which $\mathbf{x} \in \mathbb{R}^n$ is the decision variable, $u \in \mathbb{R}^m$ is the external signal, and $\theta \in \Theta$ is the parameter.

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & f(\mathbf{x}, u, \theta) \\ s.t. \quad & \mathbf{g}(\mathbf{x}, u, \theta) \leq \mathbf{0} \end{aligned} \qquad \text{DMP}$$

  where $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \mapsto \mathbb{R}$ is a real-valued function, and $\mathbf{g} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \mapsto \mathbb{R}^q$ is a vector-valued function.
- $X(u, \theta) = \{x \in \mathbb{R}^n : \mathbf{g}(\mathbf{x}, u, \theta) \leq \mathbf{0}\}$, the feasible region of DMP.
- Key concept: $S(u, \theta) = \arg\min\{f(\mathbf{x}, u, \theta) : x \in X(u, \theta)\}$, the optimal solution set of DMP.

Decision making problem

- We consider a family of parameterized optimization problem, in which $\mathbf{x} \in \mathbb{R}^n$ is the decision variable, $u \in \mathbb{R}^m$ is the external signal, and $\theta \in \Theta$ is the parameter.

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} \quad & f(\mathbf{x}, u, \theta) \\ s.t. \quad & \mathbf{g}(\mathbf{x}, u, \theta) \leq \mathbf{0} \end{aligned} \qquad \text{DMP}$$

  where $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \mapsto \mathbb{R}$ is a real-valued function, and $\mathbf{g} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \mapsto \mathbb{R}^q$ is a vector-valued function.

- $X(u, \theta) = \{x \in \mathbb{R}^n : \mathbf{g}(\mathbf{x}, u, \theta) \leq \mathbf{0}\}$, the feasible region of DMP.
- Key concept: $S(u, \theta) = \arg\min\{f(\mathbf{x}, u, \theta) : x \in X(u, \theta)\}$, the optimal solution set of DMP.
- Assumption
  Set Θ is a convex compact set.
  There exists $D > 0$ such that $\|\theta\|_2 \leq D$ for all $\theta \in \Theta$. For each $u \in \mathcal{U}, \theta \in \Theta$, both $\mathbf{f}(\mathbf{x}, u, \theta)$ and $\mathbf{g}(\mathbf{x}, u, \theta)$ are convex in $\mathbf{x}$.

Inverse optimization problem in batch setting

As in conventional studies of[3456], we consider a situation where a decision $\mathbf{y}_i$ (with respect to an external signal u_i) for $i \in [N]$ are observed and recorded, the IOP model is to infer the parameter θ in DMP by minimizing an empirical loss

$$\min_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^{N} l(\mathbf{y}_i, u_i, \theta) \qquad \text{(Batch-IOP)}$$

where $l(\mathbf{y}_i, \mathbf{u}_i, \theta)$ is a loss function that captures the discrepancy between the model inferred from data and the actual one.

[3] **bertsimas2015data**.
[4] **aswani2016inverse**.
[5] **keshavarz2011imputing**.
[6] **esfahani2017data**.

Online setting

- In our online learning setting, the signal-noisy decision pair becomes available to the learner one by one.

Online setting

- In our online learning setting, the signal-noisy decision pair becomes available to the learner one by one.
- The learning algorithm produces a sequence of hypotheses $(\theta_1, \dots, \theta_T)$.

Online setting

- In our online learning setting, the signal-noisy decision pair becomes available to the learner one by one.
- The learning algorithm produces a sequence of hypotheses $(\theta_1, \ldots, \theta_T)$.
- Let $l(\mathbf{y}_t, u_t, \theta_t)$ denote the loss the learning algorithm suffers when it tries to predict the tth decision given u_t based on $\{(u_1, \mathbf{y}_1), \cdots, (u_{t-1}, \mathbf{y}_{t-1})\}$.

Online setting

- In our online learning setting, the signal-noisy decision pair becomes available to the learner one by one.
- The learning algorithm produces a sequence of hypotheses $(\theta_1, \ldots, \theta_T)$.
- Let $l(\mathbf{y}_t, u_t, \theta_t)$ denote the loss the learning algorithm suffers when it tries to predict the tth decision given u_t based on $\{(u_1, \mathbf{y}_1), \cdots, (u_{t-1}, \mathbf{y}_{t-1})\}$.
- The goal of the learner is to minimize the regret: the cumulative loss $\sum_{t\in[T]} l(\mathbf{y}_t, u_t, \theta_t)$ against the possible loss when the whole batch of decisions are available

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

Loss function and implicit update rule

- Given a (signal,noisy decision) pair $(u, \mathbf{y})$ and a hypothesis θ, we set the loss function as the minimum (squared) distance between $\mathbf{y}$ and the optimal solution set $S(u, \theta)$ in the following.

$$l(\mathbf{y}, u, \theta) = \min_{\mathbf{x} \in S(u,\theta)} \|\mathbf{y} - \mathbf{x}\|_2^2. \tag{1}$$

Loss function and implicit update rule

- Given a (signal,noisy decision) pair $(u, \mathbf{y})$ and a hypothesis θ, we set the loss function as the minimum (squared) distance between $\mathbf{y}$ and the optimal solution set $S(u, \theta)$ in the following.

$$l(\mathbf{y}, u, \theta) = \min_{\mathbf{x} \in S(u,\theta)} \|\mathbf{y} - \mathbf{x}\|_2^2. \tag{1}$$

- Once receiving the tth (signal,noisy decision) pair $(u_t, \mathbf{y}_t)$, θ_{t+1} can be obtained by solving the following optimization problem:

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

where η_t is the learning rate in round t, and $l(\mathbf{y}_t, u_t, \theta)$ is defined in (1).

Loss function and implicit update rule

- Given a (signal,noisy decision) pair $(u, \mathbf{y})$ and a hypothesis θ, we set the loss function as the minimum (squared) distance between $\mathbf{y}$ and the optimal solution set $S(u, \theta)$ in the following.

$$l(\mathbf{y}, u, \theta) = \min_{\mathbf{x} \in S(u,\theta)} \|\mathbf{y} - \mathbf{x}\|_2^2. \tag{1}$$

- Once receiving the tth (signal,noisy decision) pair $(u_t, \mathbf{y}_t)$, θ_{t+1} can be obtained by solving the following optimization problem:

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

  where η_t is the learning rate in round t, and $l(\mathbf{y}_t, u_t, \theta)$ is defined in (1).
- Seek to balance the tradeoff between "conservativeness" and "correctiveness".

Loss function and implicit update rule

- Given a (signal,noisy decision) pair $(u, \mathbf{y})$ and a hypothesis θ, we set the loss function as the minimum (squared) distance between $\mathbf{y}$ and the optimal solution set $S(u, \theta)$ in the following.

$$l(\mathbf{y}, u, \theta) = \min_{\mathbf{x} \in S(u,\theta)} \|\mathbf{y} - \mathbf{x}\|_2^2. \tag{1}$$

- Once receiving the tth (signal,noisy decision) pair $(u_t, \mathbf{y}_t)$, θ_{t+1} can be obtained by solving the following optimization problem:

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

  where η_t is the learning rate in round t, and $l(\mathbf{y}_t, u_t, \theta)$ is defined in (1).
- Seek to balance the tradeoff between "conservativeness" and "correctiveness".
- As there is no closed form for θ_{t+1} in general, we call (2) an implicit update rule.

Algorithm

Algorithm 1 1 Implicit Online Learning for Generalized Inverse Optimization

Input: (signal,noisy decision) pairs $\{(u_t, \mathbf{y}_t)\}_{t \in [T]}$
Initialization: θ_1 could be an arbitrary hypothesis of the parameter.
for $t = 1$ to T **do**
  receive $(u_t, \mathbf{y}_t)$
  suffer loss $l(\mathbf{y}_t, u_t, \theta_t)$
  if $l(\mathbf{y}_t, u_t, \theta_t) = 0$ **then**
    $\theta_{t+1} \leftarrow \theta_t$
  else
    set learning rate $\eta_t \propto 1/\sqrt{t}$
    update $\theta_{t+1} = \arg\min_{\theta \in \Theta} \frac{1}{2}\|\theta - \theta_t\|_2^2 + \eta_t l(\mathbf{y}_t, u_t, \theta)$ (i.e., solve (2))
  end if
end for

Theoretical analysis

Theorem (Regret bound)

Suppose some technical assumptions hold. Then, choosing $\eta_t = \frac{D\lambda}{2\sqrt{2}(B+R)\kappa}\frac{1}{\sqrt{t}}$,

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

wher λ and κ are related to the smoothness of the objective functions.

Theorem (Risk consistency)

Suppose some technical assumptions hold. Then, choosing $\eta_t = \frac{D\lambda}{2\sqrt{2}(B+R)\kappa}\frac{1}{\sqrt{t}}$,

$$\frac{1}{T}\sum_{t\in[T]} l(\mathbf{y}_t, u_t, \theta_t) \xrightarrow{p} \mathbb{E}\, l(\mathbf{y}, u, \theta^*)$$

as T approaches to infinity. Here, θ^* minimizes the true risk $\mathbb{E}\ [l(\mathbf{y}, u, \theta)]$.

Learning consumer behavior

Consider the consumer's behavior problem in a market with n products. The prices for the products are denoted by $\mathbf{p}_t \in \mathbb{R}^n_+$ which varies over different time $t \in [T]$.

- The consumer's decision making problem can be stated as the following utility maximization problem (UMP)[7]

$$\begin{aligned} \max_{\mathbf{x} \in \mathbb{R}^n_+} \quad & u(\mathbf{x}) \\ s.t. \quad & \mathbf{p}_t^T \mathbf{x} \le b \end{aligned} \qquad \text{UMP}$$

  where $\mathbf{p}_t^T \mathbf{x} \le b$ is the budget constraint at time t.
- For this application, we will consider a concave quadratic representation for $u(\mathbf{x})$. That is, $u(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q \mathbf{x} + \mathbf{r}^T \mathbf{x}$, where $Q \in \mathbf{S}^n_-$ (the set of symmetric negative semidefinite matrices), $\mathbf{r} \in \mathbb{R}^n$.

[7] **mas1995microeconomic**.

Learning utility function

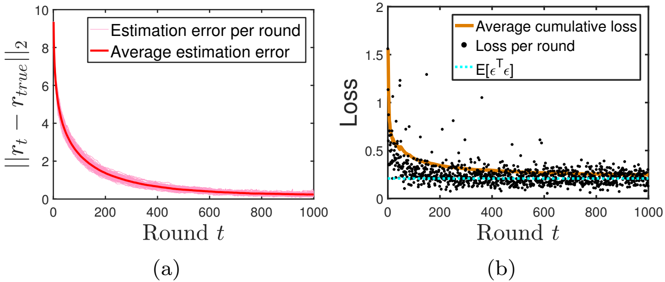
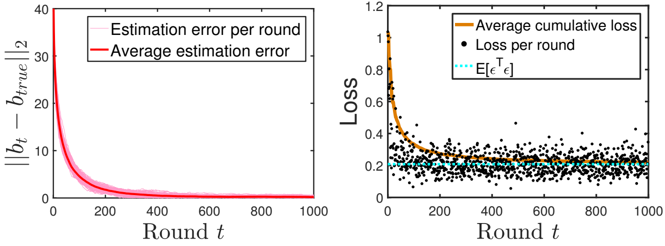


Figure: Learning the Utility Function

Learning budget



(a) (b)

Figure: Learning the Budget

Learning transportation cost

We now consider the transshipment network $G = (V_s \cup V_d, E)$, where nodes V_s are producers and the remaining nodes V_d are consumers.
Variables x_e and y_v represent transportation quantity and production quantity, respectively. The transshipment problem is

$$\begin{aligned}
\min \quad & \sum_{v \in V_s} C^v(y_v) + \sum_{e \in E} c_e x_e \\
s.t. \quad & \sum_{e \in \delta^+(v)} x_e - \sum_{e \in \delta^-(v)} x_e = y_v \quad \forall v \in V_s \\
& \sum_{e \in \delta^+(v)} x_e - \sum_{e \in \delta^-(v)} x_e = d_v^t \quad \forall v \in V_d \\
& 0 \le x_e \le u_e, \;\; 0 \le y_v \le w_v \qquad \forall e \in E, \forall v \in V_s
\end{aligned} \qquad \text{TP}$$

where we want to learn the transportation cost c_e for $e \in E$.

Learning transportation cost

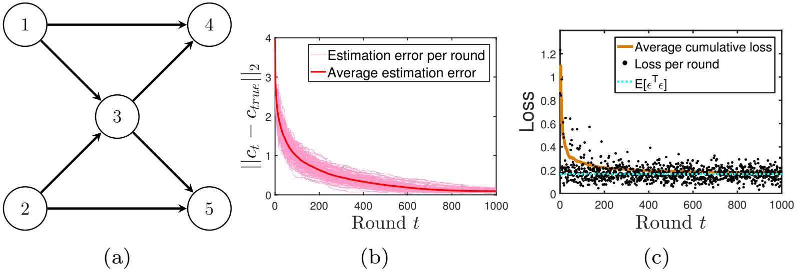


Figure: Learning the Transportation Cost

What is inverse multiobjective optimization problem (IMOP)?

- A decision is often a trade-off among multiple criteria.

What is inverse multiobjective optimization problem (IMOP)?

- A decision is often a trade-off among multiple criteria.
- Decision makers generally share similar evaluation criteria but have different priorities/preferences

What is inverse multiobjective optimization problem (IMOP)?

- A decision is often a trade-off among multiple criteria.
- Decision makers generally share similar evaluation criteria but have different priorities/preferences
- With various responses towards a same input, we are interested in (1) the common ground (i.e., multiple evaluation criteria) shared by decision makers, and
  (2) how varying they are in this decision maker population.

What is inverse multiobjective optimization problem (IMOP)?

- A decision is often a trade-off among multiple criteria.
- Decision makers generally share similar evaluation criteria but have different priorities/preferences
- With various responses towards a same input, we are interested in (1) the common ground (i.e., multiple evaluation criteria) shared by decision makers, and
  (2) how varying they are in this decision maker population.
- Given a set of observations that are noisy efficient (Pareto optimal) solutions collected from a population of decision makers, the inverse multiobjective optimization model is to infer parameter θ of a multiobjective DMP.

An example of multiobjective optimization

Consider a portfolio selection problem, where investors need to determine the fraction of their wealth to invest in each security in order to maximize the total return and minimize the total risk. The classical Markowitz mean-variance portfolio selection[8] is

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

[8] **markowitz1952portfolio**.

Pareto optimality

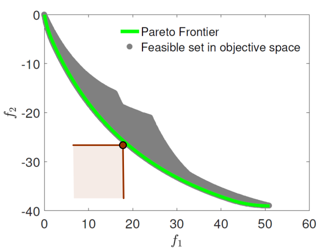
Example

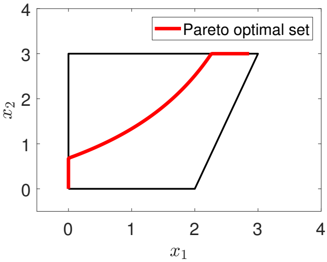
Consider the following multiobjective quadratic programming problem.

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

where parameters of the objective functions and the constraints are

$$Q_1 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, \mathbf{c}_1 = \begin{bmatrix} 3 \\ 1 \end{bmatrix}, Q_2 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, \mathbf{c}_2 = \begin{bmatrix} -6 \\ -5 \end{bmatrix}, A = \begin{bmatrix} -3 & 1 \\ 0 & -1 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} -6 \\ -3 \end{bmatrix}.$$





Pareto optimal solution

A common way to derive a Pareto optimal solution is to solve the following problem[9].

$$\begin{array}{rl} \min & w^T \mathbf{f}(\mathbf{x}, \theta) \\ s.t. & \mathbf{x} \in X(\theta) \end{array} \qquad \text{WP}$$

where $w = (w^1, \ldots, w^p)^T$ is the nonnegative weight vector in the $(p-1)$-simplex $\mathscr{W}_p \equiv \{w \in \mathbb{R}^p_+ : \mathbf{1}^T w = 1\}$.

[9] Saul et al. 1955

Motivation of our research

- Can we use inverse optimization as a surrogate for inverse multiobjective optimization?

Motivation of our research

- Can we use inverse optimization as a surrogate for inverse multiobjective optimization?
-

Motivation of our research

- Can we use inverse optimization as a surrogate for inverse multiobjective optimization?
-

Motivation of our research

- Can we use inverse optimization as a surrogate for inverse multiobjective optimization?

- Consider the bi-objective LP problem

$$\begin{aligned} \min\ & \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ s.t.\ & ax_1 + bx_2 \geq 0, \\ & bx_1 + ax_2 \geq 0, \\ & x_1 + x_2 \leq c. \end{aligned}$$

where $a > b > 0$ and $c > 0$. Right figure displays the feasible region of an instance with $a = 6, b = 1, c = 1$, i.e., the triangle AOB.

Motivation of our research

- Can we use inverse optimization as a surrogate for inverse multiobjective optimization?
- Consider the bi-objective LP problem

$$
\begin{aligned}
\min \; & \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\
s.t. \; & ax_1 + bx_2 \geq 0, \\
& bx_1 + ax_2 \geq 0, \\
& x_1 + x_2 \leq c.
\end{aligned}
$$

where $a > b > 0$ and $c > 0$. Right figure displays the feasible region of an instance with $a = 6, b = 1, c = 1$, i.e., the triangle AOB.

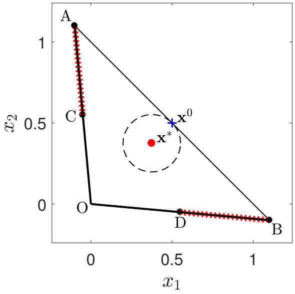


Figure: OA and OB are the efficient (solution) set for the bi-objective linear programming problem.

Motivation of our research

- Can we use inverse optimization as a surrogate for inverse multiobjective optimization?
- Consider the bi-objective LP problem

$$\min \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$
$$s.t.\ ax_1 + bx_2 \geq 0,$$
$$bx_1 + ax_2 \geq 0,$$
$$x_1 + x_2 \leq c.$$

where $a > b > 0$ and $c > 0$. Right figure displays the feasible region of an instance with $a = 6, b = 1, c = 1$, i.e., the triangle AOB.

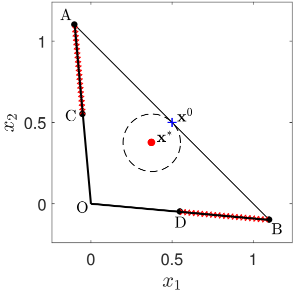
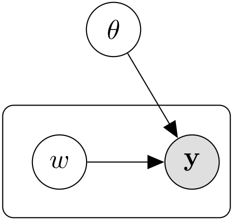


Figure: OA and OB are the efficient (solution) set for the bi-objective linear programming problem.

- The answer is NO! We would get the obj $= -x_1 - x_2$ using IOP, which reflects opposite information regarding decision makers' intentions.

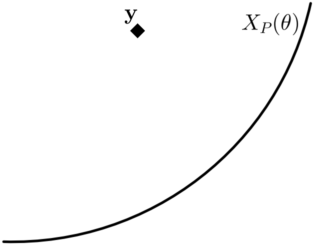
IMOP is an unsupervised learning task

The only data we have for IMOP is the noisy decisions $\{\mathbf{y}_i\}_{i \in [N]}$.



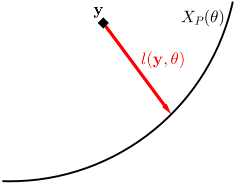
Task	# of obj	Signal	Supervised learning	Paper
IOP	single	yes	yes	(Keshavarz et al., 2011; Bertsimas et al., 2015; Aswani et al., 2018) (Esfahani et al., 2018; Bärmann et al., 2017; Dong et al., 2018)
IMOP	multiple	no	no	this paper

Loss function of unsupervised learning type



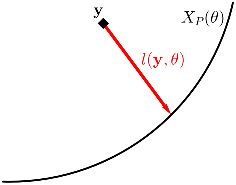
Loss function of unsupervised learning type

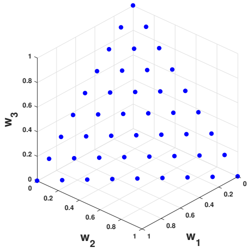
$$l(\mathbf{y}, \theta) = \min_{\mathbf{x} \in X_P(\theta)} \|\mathbf{y} - \mathbf{x}\|_2^2,$$

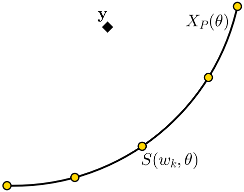


Loss function of unsupervised learning type

$$l(\mathbf{y},\theta) = \min_{\mathbf{x}\in X_P(\theta)} \|\mathbf{y}-\mathbf{x}\|_2^2,$$

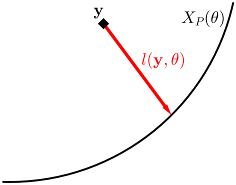


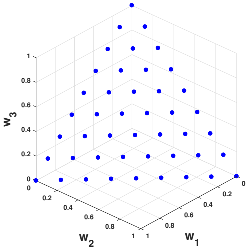


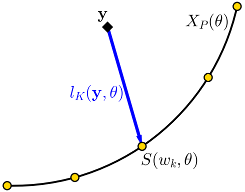


Loss function of unsupervised learning type

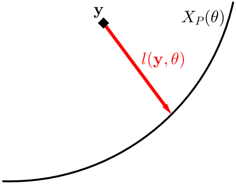
$$l(\mathbf{y}, \theta) = \min_{\mathbf{x} \in X_P(\theta)} \|\mathbf{y} - \mathbf{x}\|_2^2,$$



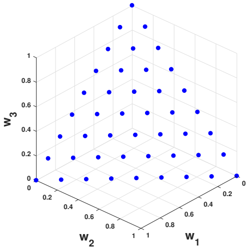


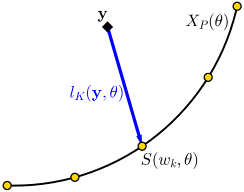


Loss function of unsupervised learning type



$$l(\mathbf{y},\theta) = \min_{\mathbf{x}\in X_P(\theta)} \|\mathbf{y}-\mathbf{x}\|_2^2,$$





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

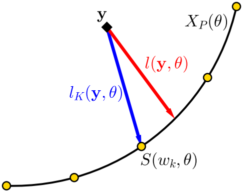
surrogate loss function

Convergence rate

Theorem

Under certein assumptions, we have that $\forall \mathbf{y} \in \mathcal{Y}, \forall \theta \in \Theta$,

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



Example

When $p = 2$, i.e., a bi-objective decision making problem, theorem shows that $l_K(\mathbf{y}, \theta) - l(\mathbf{y}, \theta)$ is of $\mathcal{O}(1/K)$.

Model for IMOP

- Given a set of observations that are noisy efficient solutions $\{\mathbf{y}_i\}_{i \in [N]}$, construct an optimization model to infer parameter θ of a multiobjective DMP.

[10] **dong2018inferring**.

Model for IMOP

- Given a set of observations that are noisy efficient solutions $\{\mathbf{y}_i\}_{i\in[N]}$, construct an optimization model to infer parameter θ of a multiobjective DMP.
- Loss function. We adopt a sampling approach to generate weights $w_k \in \mathscr{W}_p$ for each $k \in [K]$ and approximate $X_E(\theta)$ as the union of their $S(w_k, \theta)$s. Then, by utilizing binary variables, the loss function is

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

[10] **dong2018inferring**.

Model for IMOP

- Given a set of observations that are noisy efficient solutions $\{\mathbf{y}_i\}_{i\in[N]}$, construct an optimization model to infer parameter θ of a multiobjective DMP.
- Loss function. We adopt a sampling approach to generate weights $w_k \in \mathscr{W}_p$ for each $k \in [K]$ and approximate $X_E(\theta)$ as the union of their $S(w_k, \theta)$s. Then, by utilizing binary variables, the loss function is

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

- Model for IMOP[10]

$$
\begin{aligned}
\min_{\theta\in\Theta} \quad & M_K^N(\theta) \equiv \frac{1}{N} \sum_{i\in[N]} \|\mathbf{y}_i - \sum_{k\in[K]} z_{ik} \mathbf{x}_k\|_2^2 \\
\text{s.t.} \quad & \mathbf{x}_k \in S(w_k, \theta), && \forall k \in [K], \\
& \sum_{k\in[K]} z_{ik} = 1, && \forall i \in [N], \\
& z_{ik} \in \{0, 1\}, && \forall i \in [N], \; k \in [K].
\end{aligned}
\tag{IMOP}
$$

[10] **dong2018inferring**.

Statistical properties of IMOP

Theorem (Consistency of IMOP)

$$
\begin{tikzcd}[column sep=huge, row sep=huge]
M_K^N(\theta) \arrow[r, "P"] \arrow[d, "P"'] \arrow[dr, dashed, "P"] & M^N(\theta) \arrow[d, "P"] \\
M_K(\theta) \arrow[r] & M(\theta)
\end{tikzcd}
$$

Figure: Uniform convergence diagram for empirical risks. $\xrightarrow{P}$ means convergence in probability. $\longrightarrow$ indicates the convergence of a sequence of numbers. $\overset{P}{\dashrightarrow}$ means convergence in probability for double-index random variable.

K-means Clustering

K-means clustering aims to partition the observations into K clusters (or groups) such that the average squared distance between each observation and its closest cluster centroid is minimized. Given observations $\{\mathbf{y}_i\}_{i\in[N]}$[11],

$$\begin{aligned}
\min_{\mathbf{x}_k, z_{ik}} \quad & \frac{1}{N}\sum_{i\in[N]} \|\mathbf{y}_i - \sum_{k\in[K]} z_{ik}\mathbf{x}_k\|_2^2 & \\
\text{s.t.} \quad & \sum_{k\in[K]} z_{ik} = 1, & \forall i \in [N], \\
& \mathbf{x}_k \in \mathbb{R}^n, \;\; z_{ik} \in \{0,1\}, & \forall i \in [N],\; k \in [K],
\end{aligned}$$

K-means clustering

where K is the number of clusters, and $\{\mathbf{x}_k\}_{k\in[K]}$ are the centroids of the clusters.

[11] **bagirov2008modified**; **aloise2009branch**.

Connection between IMOP and Clustering

Theorem

Given any K-means clustering problem, we can construct an instance of IMOP, such that solving K-means clustering is equivalent to solving the IMOP.

The key to prove the theorem:

$$\begin{aligned} \min_{\theta \in \Theta} \quad & M_K^N(\theta) \equiv \frac{1}{N} \sum_{i \in [N]} \|\mathbf{y}_i - \sum_{k \in [K]} z_{ik} \mathbf{x}_k\|_2^2 \\ \text{s.t.} \quad & \mathbf{x}_k \in S(w_k, \theta), \\ & \sum_{k \in [K]} z_{ik} = 1, \\ & z_{ik} \in \{0, 1\}. \end{aligned}$$

$$\begin{aligned} \min_{\mathbf{x}_k, z_{ik}} \quad & \frac{1}{N} \sum_{i \in [N]} \|\mathbf{y}_i - \sum_{k \in [K]} z_{ik} \mathbf{x}_k\|_2^2 \\ \text{s.t.} \quad & \mathbf{x}_k \in \mathbb{R}^n, \\ & \sum_{k \in [K]} z_{ik} = 1, \\ & z_{ik} \in \{0, 1\}. \end{aligned}$$

where K is the number of clusters, and $\{\mathbf{x}_k\}_{k \in [K]}$ are the centroids.

Complexity of IMOP

Lemma (NP-hardness of K-means clustering)

K–means clustering is NP-hard to solve even for instances in the plane[12]*, or with two clusters in the general dimension*[13]*.*

Theorem (NP-hardness of IMOP)

IMOP is NP-hard to solve.

[12] Meena et al. 2012
[13] Daniel et al. 2009

Connection between IMOP and Clustering

- Clearly, in both IMOP and K-means clustering, one needs to assign $\{\mathbf{y}_i\}_{i\in[N]}$ to certain clusters in such a way that the average squared distance between $\mathbf{y}_i$ and its closest $\mathbf{x}_k$ is minimized.
- The difference is whether $\mathbf{x}_k$ has restriction or not. In IMOP, each $\mathbf{x}_k$ is restricted to belong to $S(w_k,\theta)$, while there is no restriction for $\mathbf{x}_k$ in K-means clustering.
- We partition $\{\mathbf{y}_i\}_{i\in[N]}$ into K clusters $\{C_k\}_{k\in[K]}$. Let $\overline{\mathbf{y}}_k = \frac{1}{|C_k|}\sum_{\mathbf{y}_i\in C_k}\mathbf{y}_i$ be the centroid of cluster C_k.

$$
\begin{aligned}
\min_{\theta,\mathbf{x}_{k'}} \quad & \frac{1}{N}\sum_{k\in[K]} |C_k|\|\overline{\mathbf{y}}_k - \textstyle\sum_{k'\in[K]} z_{kk'}\mathbf{x}_{k'}\|_2^2 \\
\text{s.t.} \quad & \mathbf{x}_{k'} \in S(w_{k'},\theta), && \forall k' \in [K], \\
& \sum_{k'\in[K]} z_{kk'} = 1, && \forall k \in [K], \\
& z_{kk'} \in \{0,1\}, && \forall k \in [K],\ k' \in [K].
\end{aligned}
$$

Kmeans-IMOP

Algorithm

Algorithm 2 2 Solving IMOP through a Clustering-based Approach

1: **Input:** Noisy decisions $\{\mathbf{y}_i\}_{i\in[N]}$, weight samples $\{w_k\}_{k\in[K]}$.

2: **Initialize** Partition $\{\mathbf{y}_i\}_{i\in[N]}$ into K clusters using K-means clustering. Calculate $\{\overline{\mathbf{y}}_k\}_{k\in[K]}$. Solve Kmeans-IMOP and get an initial estimation of θ and $\{\mathbf{x}_k\}_{k\in[K]}$.

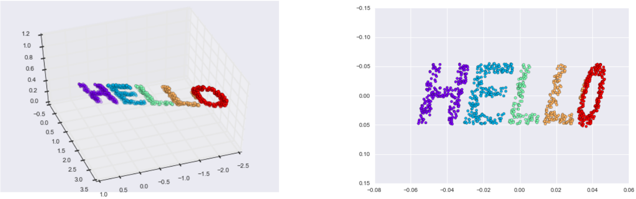
3: **while** stopping criterion is not satisfied **do**

4: **Assignment step**: Assign each $\mathbf{y}_i$ to the closest $\mathbf{x}_k$ to form new clusters. Calculate their centroids $\{\overline{\mathbf{y}}_k\}_{k\in[K]}$.

5: **Update step**: Update θ and $\{\mathbf{x}_k\}_{k\in[K]}$ by solving Kmeans-IMOP.

6: **end while**

Manifold learning



(a) High dimension space

(b) Low dimension space

Formally, given a set of high-dimensional data points $\{\mathbf{y}_i\}_{i\in[N]}$ in $\mathbb{R}^n$, we are required to find a mapping $f : \mathbb{R}^d \to \mathbb{R}^n$ and an embedding set $\{\mathbf{x}_i\}_{i\in[N]}$ in a low-dimensional space $\mathbb{R}^d$ $(d < n)$ such that

$$\mathbf{y}_i = f(\mathbf{x}_i) + \epsilon_i, \quad i \in [N],$$

and the local manifold structure formed by $\{\mathbf{y}_i\}_{i\in[N]}$ is preserved in the embedded space[14]. Here, ϵ_i represents random noise.

[14] Joshua et al. 2000; Sam et al. 2000

Pareto manifold

Theorem (Pareto manifold)

Suppose certain regularity assumptions hold. For each $\theta \in \Theta$, the Pareto optimal set is a $(p-1)$-dimensional piecewise continuous manifold.

Corollary

Suppose that both $\mathbf{f}(\mathbf{x}, \theta)$ and $\mathbf{g}(\mathbf{x}, \theta)$ are linear functions in $\mathbf{x}$ for all $\theta \in \Theta$. Then, $X_P(\theta)$ is a $(p-1)$-dimensional piecewise linear manifold for all $\theta \in \Theta$.

Pareto manifold

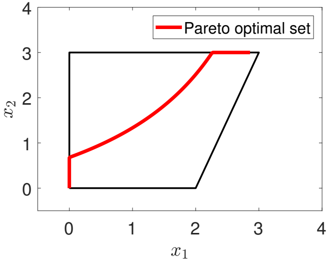
Theorem (Pareto manifold)

Suppose certain regularity assumptions hold. For each $\theta \in \Theta$, the Pareto optimal set is a $(p-1)$-dimensional piecewise continuous manifold.

Corollary

Suppose that both $\mathbf{f}(\mathbf{x}, \theta)$ and $\mathbf{g}(\mathbf{x}, \theta)$ are linear functions in $\mathbf{x}$ for all $\theta \in \Theta$. Then, $X_P(\theta)$ is a $(p-1)$-dimensional piecewise linear manifold for all $\theta \in \Theta$.

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



Pareto manifold

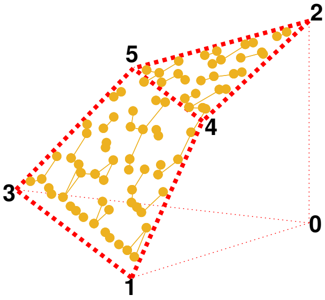
Theorem (Pareto manifold)

Suppose certain regularity assumptions hold. For each $\theta \in \Theta$, the Pareto optimal set is a $(p-1)$-dimensional piecewise continuous manifold.

Corollary

Suppose that both $\mathbf{f}(\mathbf{x}, \theta)$ and $\mathbf{g}(\mathbf{x}, \theta)$ are linear functions in $\mathbf{x}$ for all $\theta \in \Theta$. Then, $X_P(\theta)$ is a $(p-1)$-dimensional piecewise linear manifold for all $\theta \in \Theta$.

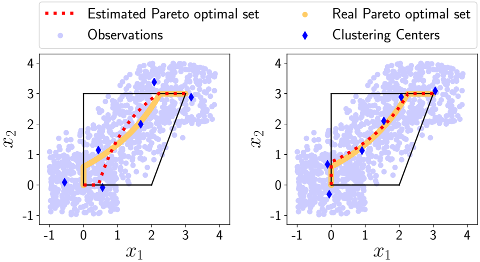
$$\begin{array}{rl} \min & \{-x_1, -x_2, -x_3\} \\ s.t. & x_1 + x_2 + x_3 \leq 5, \\ & x_1 + x_2 + 3x_3 \leq 9, \\ & x_1, x_2, x_3 \geq 0. \end{array}$$



An enhanced algorithm with manifold learning

Algorithm 3 3 Initialization with manifold learning

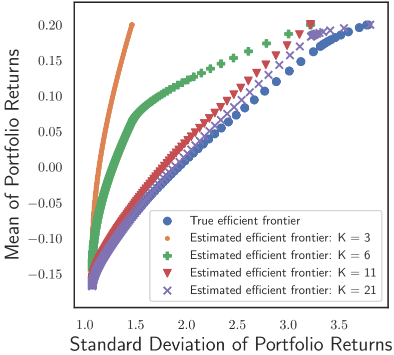
1: **Input:** Noisy decision $\{\mathbf{y}_i\}_{i\in[N]}$, evenly sampled weights $\{w_k\}_{k\in[K]}$.
2: Apply any nonlinear manifold learning algorithm: $\mathbf{y}_i \in \mathbb{R}^n \to \mathbf{x}_i \in \mathbb{R}^{p-1}, \forall i \in [N]$.
3: Group $\{\mathbf{x}_i\}_{i\in[N]}$ into K clusters by K-means clustering.
Denote I_K the set of labels of $\{\mathbf{x}_i\}_{i\in[N]}$.
Find the clusters $\{C_k\}_{k\in[K]}$ and centroids $\{\overline{\mathbf{y}}_k\}_{k\in[K]}$ of $\{\mathbf{y}_i\}_{i\in[N]}$ according to I_K.
4: Solve (Kmeans-IMOP) and get $\hat{\theta}$ and $\{\mathbf{x}_k\}_{k\in[K]}$.
5: Run Step 3 - 6 in Algorithm 2.



Learning the expected returns

The classical Markovitz mean-variance portfolio selection in the following is often used by analysts.

$$
\begin{aligned}
\min_{\mathbf{x}} \quad & \begin{pmatrix} f_1(\mathbf{x}) & = -\mathbf{r}^T\mathbf{x} \\ f_2(\mathbf{x}) & = \mathbf{x}^T Q\mathbf{x} \end{pmatrix} \\
s.t. \quad & 0 \le x_i \le b_i, \qquad \forall i \in [n], \\
& \sum_{i=1}^{n} x_i = 1,
\end{aligned}
$$



Inverse optimization

- Propose the first general framework for eliciting decision maker's preferences and restrictions using inverse optimization through online learning.

Inverse optimization

- Propose the first general framework for eliciting decision maker's preferences and restrictions using inverse optimization through online learning.
- Learn general convex utility functions and restrictions with observed noisy signal-decision pairs.

Inverse optimization

- Propose the first general framework for eliciting decision maker's preferences and restrictions using inverse optimization through online learning.
- Learn general convex utility functions and restrictions with observed noisy signal-decision pairs.
- Prove that the online learning algorithm has a $\mathcal{O}(\sqrt{T})$ regret under certain regularity conditions. Hence, this method has a fast convergence rate.

Inverse optimization

- Propose the first general framework for eliciting decision maker's preferences and restrictions using inverse optimization through online learning.
- Learn general convex utility functions and restrictions with observed noisy signal-decision pairs.
- Prove that the online learning algorithm has a $\mathcal{O}(\sqrt{T})$ regret under certain regularity conditions. Hence, this method has a fast convergence rate.
- The algorithm can learn the parameters with great accuracy and is very robust to noises, and achieves drastic improvement in computational efficiency over the batch learning approach.

Inverse optimization

- Propose the first general framework for eliciting decision maker's preferences and restrictions using inverse optimization through online learning.
- Learn general convex utility functions and restrictions with observed noisy signal-decision pairs.
- Prove that the online learning algorithm has a $\mathcal{O}(\sqrt{T})$ regret under certain regularity conditions. Hence, this method has a fast convergence rate.
- The algorithm can learn the parameters with great accuracy and is very robust to noises, and achieves drastic improvement in computational efficiency over the batch learning approach.
- Future work for the inverse optimization will mainly focus on the application of the online learning methods, e.g., in designing recommender systems.

Inverse multiobjective optimization

- Develop a new inverse multiobjective optimization problem (IMOP) that is able to infer multiple criteria (or constraints) over which the Pareto optimal decisions are made.

Inverse multiobjective optimization

- Develop a new inverse multiobjective optimization problem (IMOP) that is able to infer multiple criteria (or constraints) over which the Pareto optimal decisions are made.
- Provide a solid analysis to ensure the statistical significance of the inference results from our IMOP model.

Inverse multiobjective optimization

- Develop a new inverse multiobjective optimization problem (IMOP) that is able to infer multiple criteria (or constraints) over which the Pareto optimal decisions are made.
- Provide a solid analysis to ensure the statistical significance of the inference results from our IMOP model.
- Reveal a hidden connection between our IMOP and the K-means clustering problem, and leverage the connection and its manifold structure in designing powerful algorithms to handle many noisy data.

Inverse multiobjective optimization

- Develop a new inverse multiobjective optimization problem (IMOP) that is able to infer multiple criteria (or constraints) over which the Pareto optimal decisions are made.
- Provide a solid analysis to ensure the statistical significance of the inference results from our IMOP model.
- Reveal a hidden connection between our IMOP and the K-means clustering problem, and leverage the connection and its manifold structure in designing powerful algorithms to handle many noisy data.
- On-going work

Inverse multiobjective optimization

- Develop a new inverse multiobjective optimization problem (IMOP) that is able to infer multiple criteria (or constraints) over which the Pareto optimal decisions are made.
- Provide a solid analysis to ensure the statistical significance of the inference results from our IMOP model.
- Reveal a hidden connection between our IMOP and the K-means clustering problem, and leverage the connection and its manifold structure in designing powerful algorithms to handle many noisy data.
- On-going work
  - Developing online learning algorithms for IMOP

Inverse multiobjective optimization

- Develop a new inverse multiobjective optimization problem (IMOP) that is able to infer multiple criteria (or constraints) over which the Pareto optimal decisions are made.
- Provide a solid analysis to ensure the statistical significance of the inference results from our IMOP model.
- Reveal a hidden connection between our IMOP and the K-means clustering problem, and leverage the connection and its manifold structure in designing powerful algorithms to handle many noisy data.
- On-going work
  - Developing online learning algorithms for IMOP
  - Investigation of the robustness of IMOP

Inverse multiobjective optimization

- Develop a new inverse multiobjective optimization problem (IMOP) that is able to infer multiple criteria (or constraints) over which the Pareto optimal decisions are made.
- Provide a solid analysis to ensure the statistical significance of the inference results from our IMOP model.
- Reveal a hidden connection between our IMOP and the K-means clustering problem, and leverage the connection and its manifold structure in designing powerful algorithms to handle many noisy data.
- On-going work
  - Developing online learning algorithms for IMOP
  - Investigation of the robustness of IMOP
  - Applications to address real problems: Dong and Vanguard's work on learning time varying risk preferences from investment portfolios

Thank you! Any question?

Contact: bzeng@pitt.edu