CPAIOR 2019

Deep Inverse Optimization

Yingcong Tan [1], Andrew Delong [2], Daria Terekhov [1]

[1]Department of Mechanical, Industrial and Aerospace Engineering
Concordia University

[2]Department of Computer Science and Software Engineering
Concordia University

Thursday, June 6th, 2019

Agenda

I. Motivation

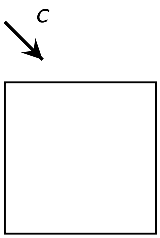
II. Methodology

III. Experiments

IV. Summary

What is Inverse Optimization (IO)?

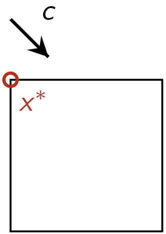
Forward Optimization Problem



$$\begin{array}{ll} \min_{x} & c'x \\ \text{s.t.} & Ax \leq b \end{array}$$

What is Inverse Optimization (IO)?

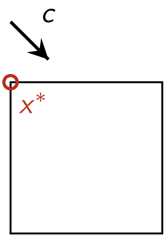
Forward Optimization Problem



$$\begin{aligned} \min_{\mathbf{x}} \quad & c'x \\ \text{s.t.} \quad & Ax \leq b \end{aligned}$$

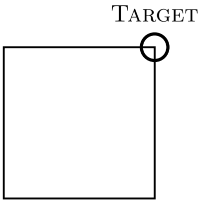
What is Inverse Optimization (IO)?

Forward Optimization Problem



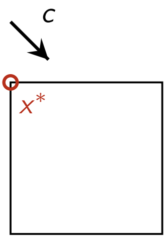
$$\begin{array}{ll} \min_{\mathbf{x}} & c'x \\ \text{s.t.} & Ax \leq b \end{array}$$

Inverse Optimization Problem



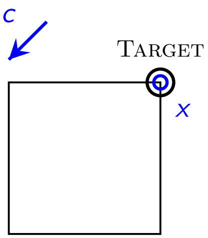
What is Inverse Optimization (IO)?

Forward Optimization Problem



$$\begin{array}{ll} \min_{\mathbf{x}} & c'x \\ \text{s.t.} & Ax \leq b \end{array}$$

Inverse Optimization Problem



$$\begin{array}{ll} \min_{\mathbf{c}} & \|\textsc{Target}, x\| \\ \text{s.t.} & x \in \arg\min_x \{c'x | Ax \leq b\} \end{array}$$

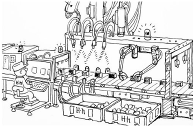
Motivation

Routing Problem (i.e., Least Cost Path)

Objective Learn the arc cost

Production Planning Problem

Objective Estimate backorder cost



Customer Behavior

Objective Estimate customer utility function

Contribution

Existing algorithms

Highlights	Optimization formulations based on optimality conditions
	Guarantee optimal solution
Limitation	Algorithms are tailored to solve special cases of IO problems
	*Chan et al. [2–4], Troutt et al.[6, 7], Aswani et al. [1], Saez-Gallego and Morales [5]

Contribution

Existing algorithms

Highlights Optimization formulations based on optimality conditions

Guarantee optimal solution

Limitation Algorithms are tailored to solve special cases of IO problems

*Chan et al. [2–4], Troutt et al.[6, 7], Aswani et al. [1], Saez-Gallego and Morales [5]

Deep Inverse Optimization

Highlights First deep-learning based approach

Learn parameters through backpropogation

Generally applicable to different IO problems

Limitation Doesn't guarantee optimal solution

Agenda

I. Motivation

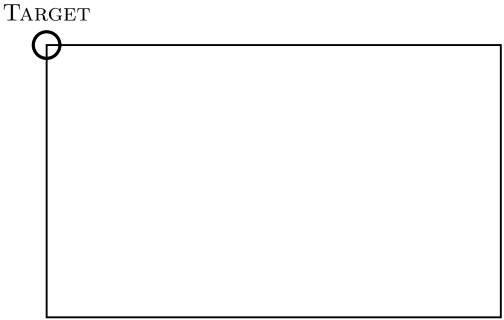
II. Methodology

III. Experiments

IV. Summary

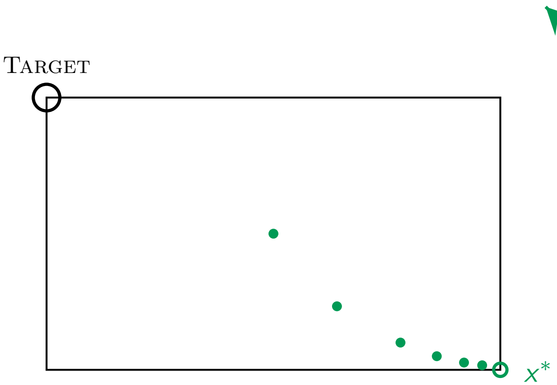
Methodology

(IO): Find a Cost Vector Consistent With Target



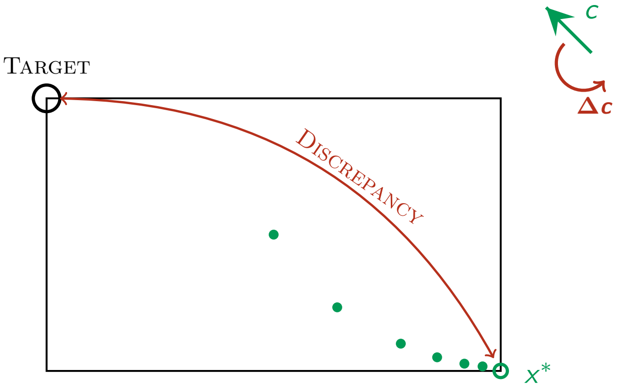
Methodology

Solve FOP using Interior-Point Method (IPM)



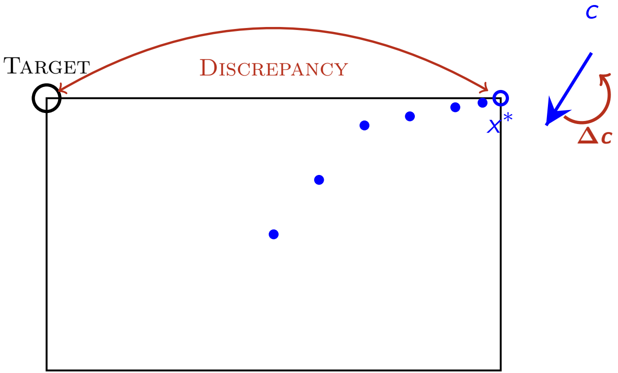
Methodology

Observe Discrepancy and Compute Gradients



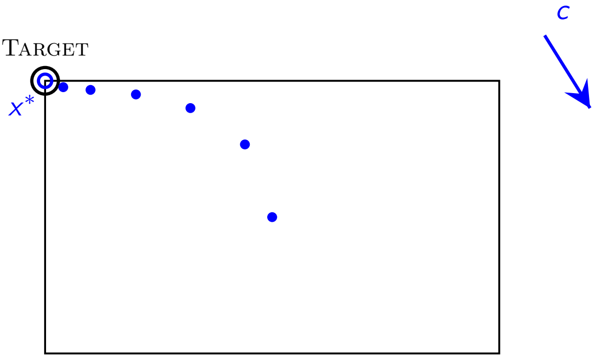
Methodology

Observe Discrepancy and Compute Gradients



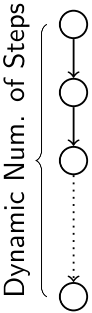
METHODOLOGY

TERMINATION



Methodology

Unroll a Deep RNN



$x^{(0)} = \textit{features}$

$x^{(1)} = \mathrm{RNN}(x^{(0)}, \textit{weights})$

$x^{(2)} = \mathrm{RNN}(x^{(1)}, \textit{weights})$

$x^{(n)} = \mathrm{RNN}(x^{(n-1)}, \textit{weights})$

Backpropogation

$$\frac{\partial\ \textsc{Loss}}{\partial\ \textit{weights}}$$

$$\min_{\textit{weights}}\ \textsc{Loss}(\textsc{Target}, x^{(n)})$$

Methodology

Unroll a Deep RNN

Dynamic Num. of Steps

$$x^{(0)} = \textit{features}$$

$$x^{(1)} = \text{RNN}(x^{(0)}, \textit{weights})$$

$$x^{(2)} = \text{RNN}(x^{(1)}, \textit{weights})$$

$$x^{(n)} = \text{RNN}(x^{(n-1)}, \textit{weights})$$

$$\min_{\textit{weights}} \ \textsc{Loss}(\textsc{Target}, x^{(n)})$$

Deep Inverse Optimization

Unroll the IPM

$$c, A, b = \text{DefineLP}(\textit{features}, \textit{weights})$$

$$x^{(0)} = \text{FindFeasible}(c, A, b)$$

$$x^{(1)} = \text{Newton}(x^{(0)}, c, A, b)$$

$$x^{(2)} = \text{Newton}(x^{(1)}, c, A, b)$$

$$x^{(n)} = \text{Newton}(x^{(n-1)}, c, A, b)$$

$$\min_{\textit{weights}} \ \textsc{Loss}(\textsc{Target}, x^{(n)})$$

Agenda

Experiments on Three Learning Tasks

Task 1 Single-point non-parametric LP

Goal Learn cost vector

- Closed-form solution proposed by Chan et al. [2, 4]

Experiments on Three Learning Tasks

Task 1 Single-point non-parametric LP

Goal Learn cost vector

- Closed-form solution proposed by Chan et al. [2, 4]

Task 2 Single-point non-parametric LP

Goal Learn cost vector and constraints jointly

- Maximum likelihood estimation approach proposed by Troutt et al. [6]

Experiments on Three Learning Tasks

Task 1 Single-point non-parametric LP

Goal Learn cost vector

- Closed-form solution proposed by Chan et al. [2, 4]

Task 2 Single-point non-parametric LP

Goal Learn cost vector and constraints jointly

- Maximum likelihood estimation apprach proposed by Troutt et al. [6]

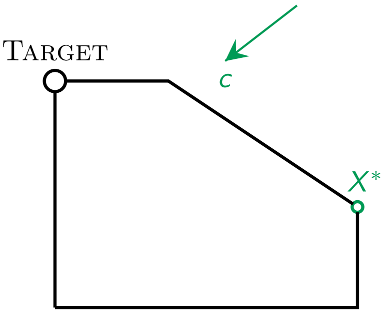
Task 3 Multi-point parametric LP, i.e., $c, A, b = f(features, weights)$

Goal Learn weights

- Not addressed in literature

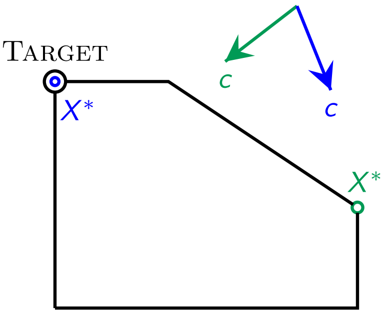
Experiment on Task 1

Goal Learn cost vector consistent with a single observed target



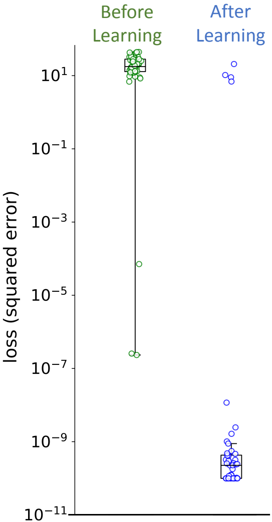
Experiment on Task 1

Goal Learn cost vector consistent with a single observed target

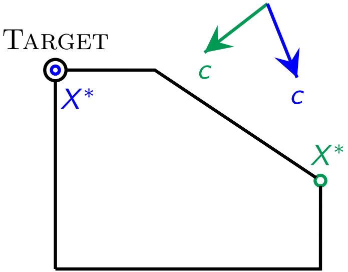


- Test on 300 random LP instances

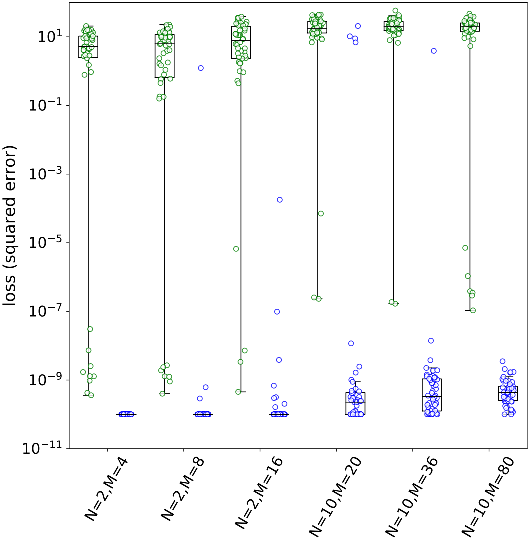
Experiment on Task 1



N = 10 Variables
M = 20 Constraints

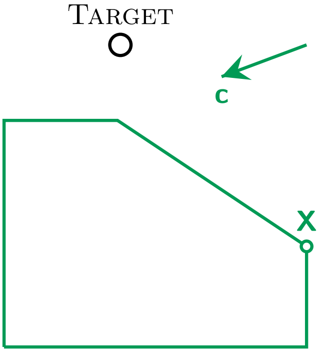


Squared Error (Learned vs Target)



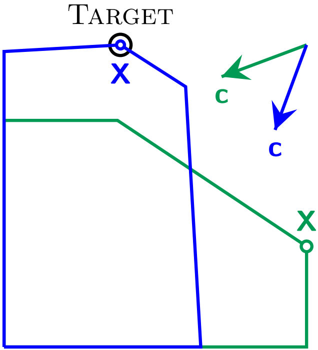
Experiment on Task 2

Goal Learn cost vector and constraints consistent with a single observed target



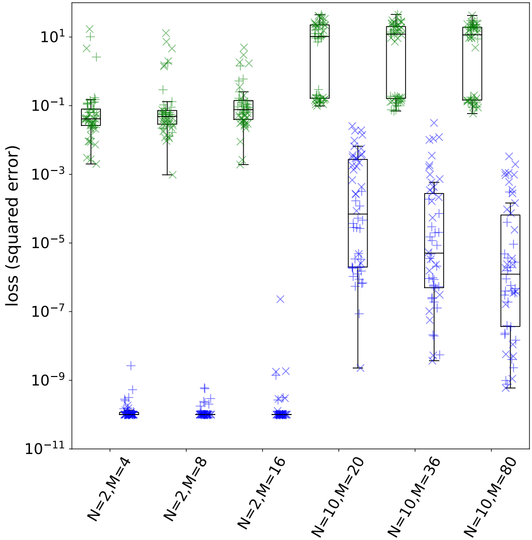
Experiment on Task 2

Goal Learn cost vector and constraints consistent with a single observed target



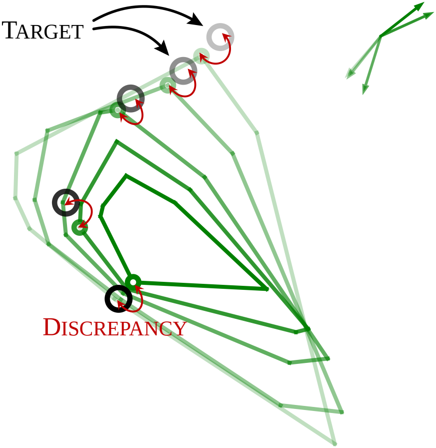
- Test on 300 random LP instances

Squared Error (Learned vs Target)



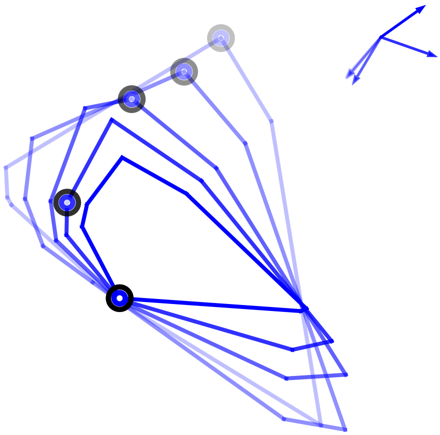
Experiment on Task 3

Goal Learn weights such that decisions are consistent with observed targets across multiple conditions

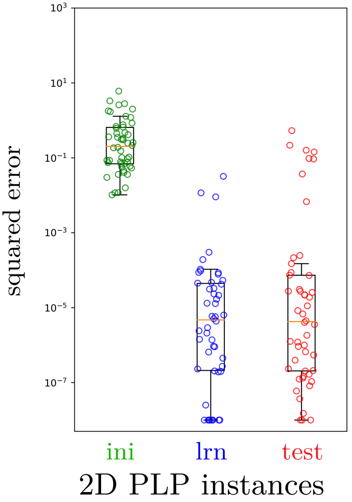


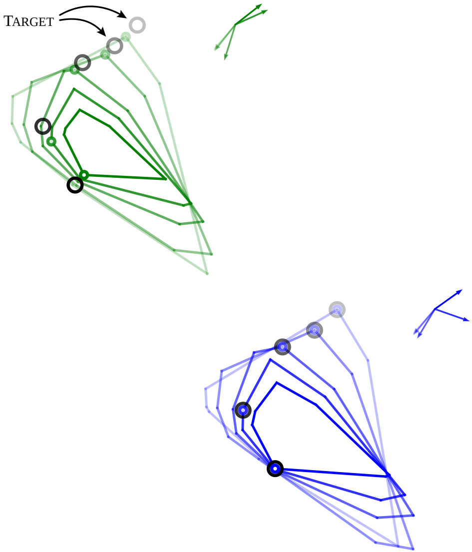
Experiment on Task 3

Goal Learn weights such that decisions are consistent with observed targets across multiple conditions

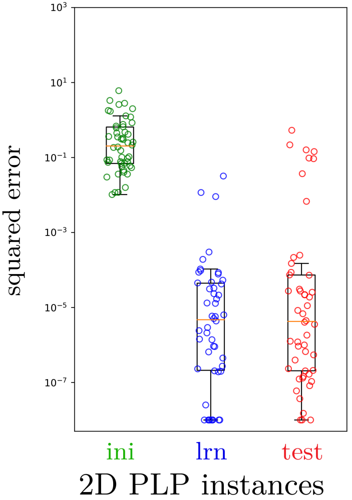


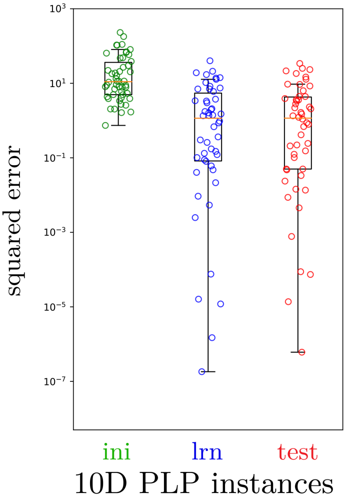
Squared Error (Learned vs Target)





Squared Error (Learned vs Target)





Agenda

I. Motivation

II. Methodology

III. Experiments

IV. Summary

Summary

- **General-purpose framework** for solving IO problems
  - Solves parametric or non-parametric problems
  - Learns all parameters individually or jointly
  - Easily extends to non-linear problems
- **Deep-Inv-Opt** package is now available on https://github.com/tankconcordia/deep_inv_opt

Thank you !