Data-Driven Optimization Technique (Inverse Optimization) for Recommender Systems

Mahsa Moghaddass, Ph.D.

April, 2020

Contents

Problem Statement

Methodology

Data & Experiments

Conclusion

Contents

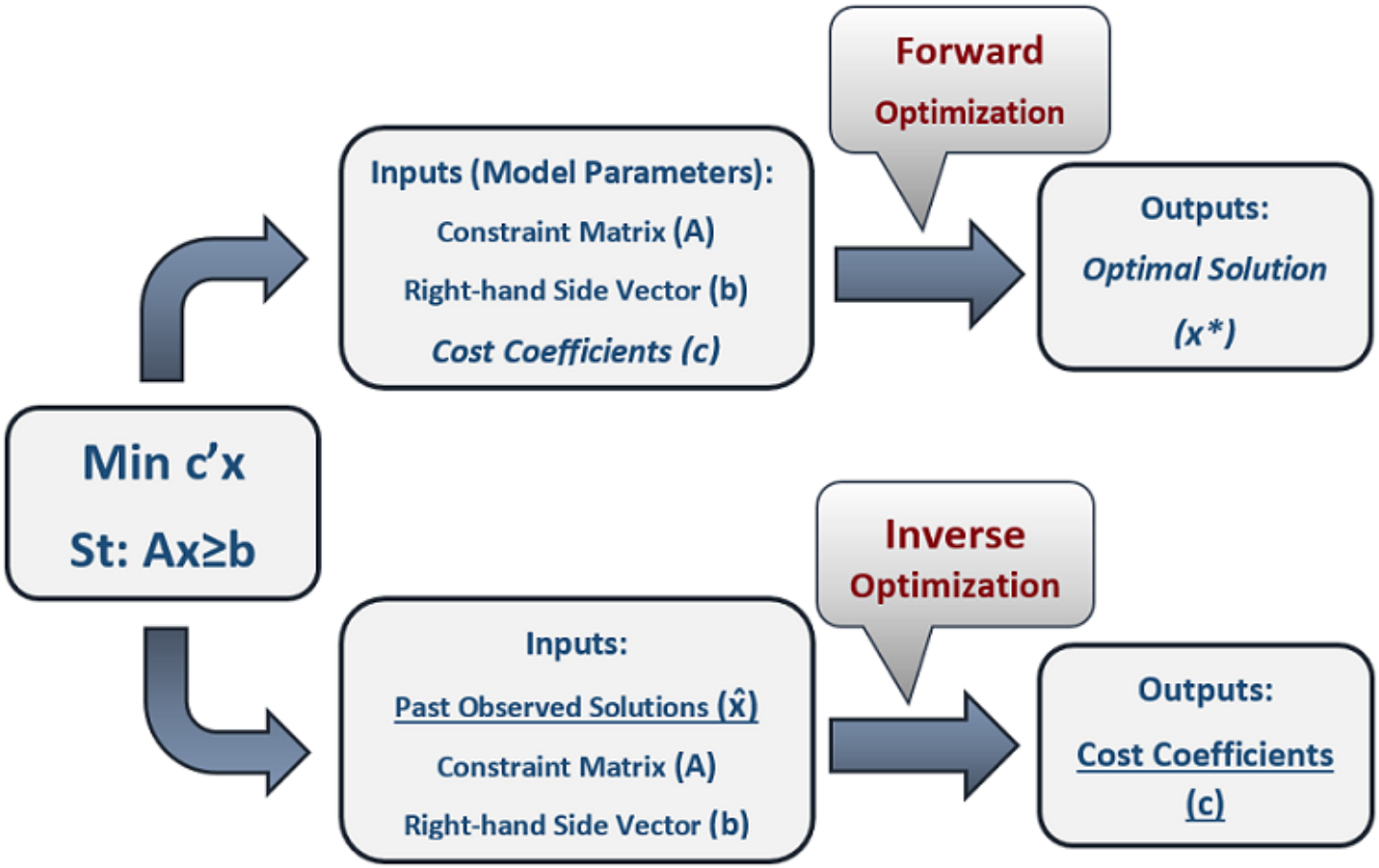
Problem Statement

Methodology

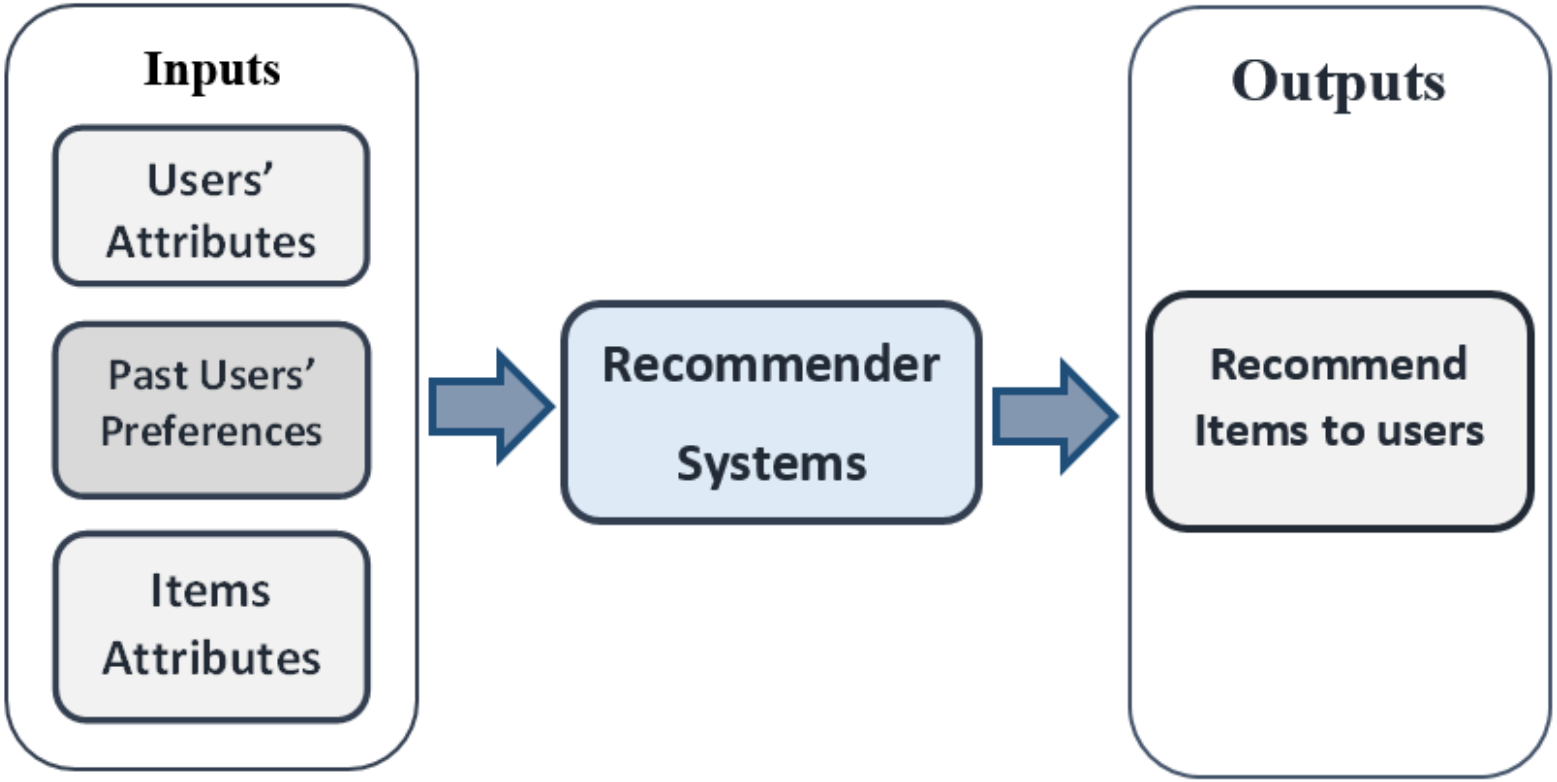
Data & Experiments

Conclusion

What is Inverse Optimization (IO)?



Novel Application in Recommender Systems



Motivation

- We **personalize** the customer' recommendations.
- Recommending items of interests will:
  - Increase **Customer Engagement** and satisfaction
  - Increase **Business** profits and bring **Values** to the system.

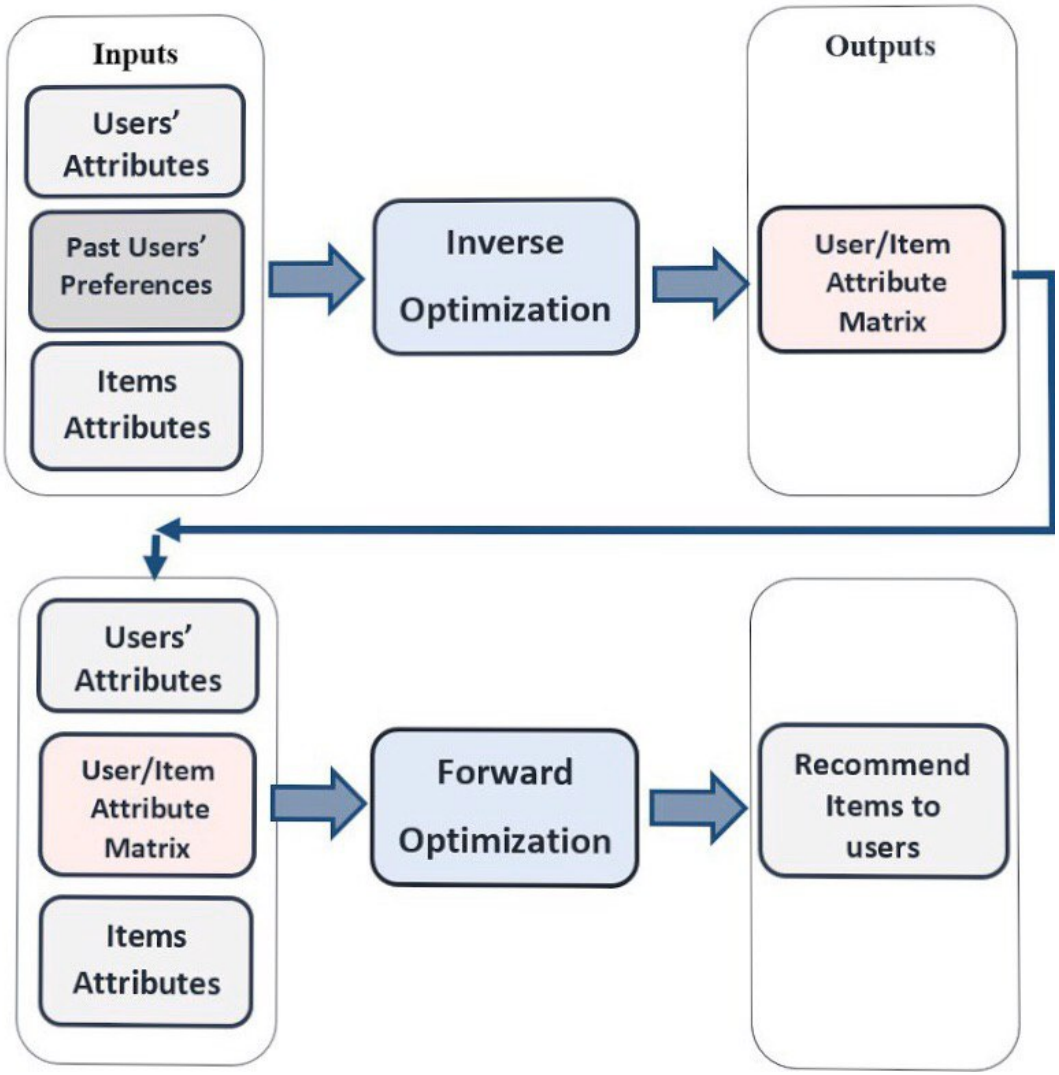
Contents

Problem Statement

Methodology

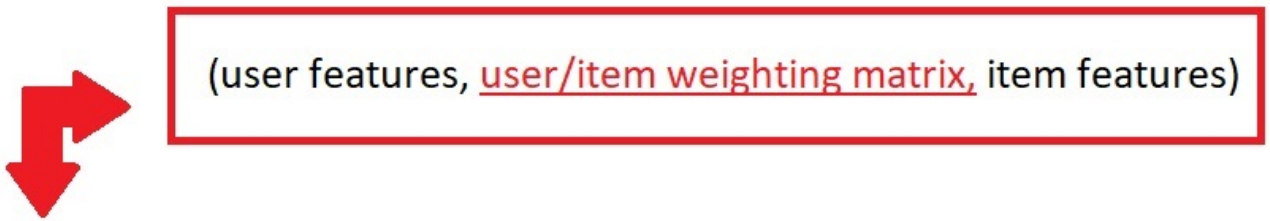
Data & Experiments

Conclusion


Inputs
Users' Attributes
Past Users' Preferences
Items Attributes
Inverse Optimization
Outputs
User/Item Attribute Matrix
Users' Attributes
User/Item Attribute Matrix
Items Attributes
Forward Optimization
Recommend Items to users

Optimization Problem

- **Objective:** Selects the **combination of items** that **maximizes** the **total values** of the **chosen items**.
- **Prediction model:**



Maximize: (weight) * user-items ratings

S.t: Sum (user-items ratings (1)) ≤ # of Demand(items)

user-items ratings ∈ {0,1} ← **Recommending items of interests**

Inverse Optimization finds user/item weighting matrix

Contents

Problem Statement

Methodology

Data & Experiments

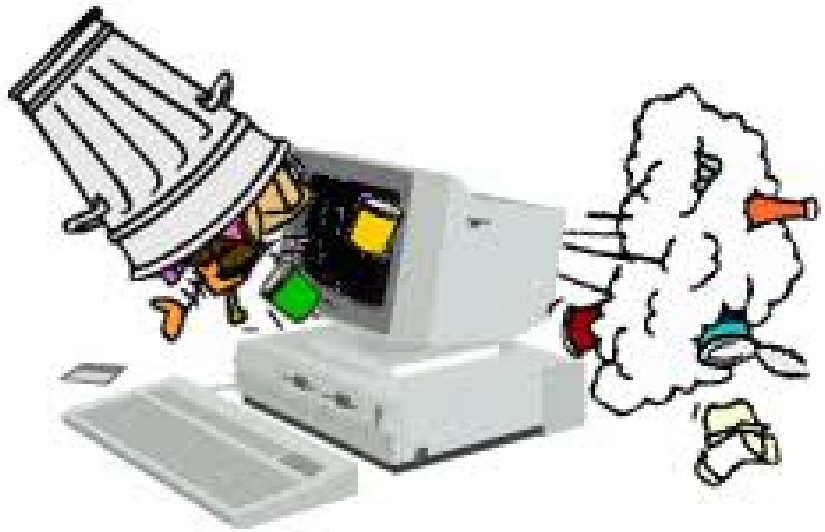
Conclusion

Data Analysis

- **Total # of users** = 116
- **User Attributes (45):** Smoking, Drink level, Dress preferences, Budget, Payment methods, etc.
- **Total # of Restaurants** = 130
- **Item Attributes (33):** Alcohol area, Smoking area, Dress code, Price, Parking area,etc.
- **Training** finds **matrices**.
- **Testing** uses the matrices in the unused data for prediction.
- The **averages** of **predicted ratings** will be compared to the **actual ratings**.

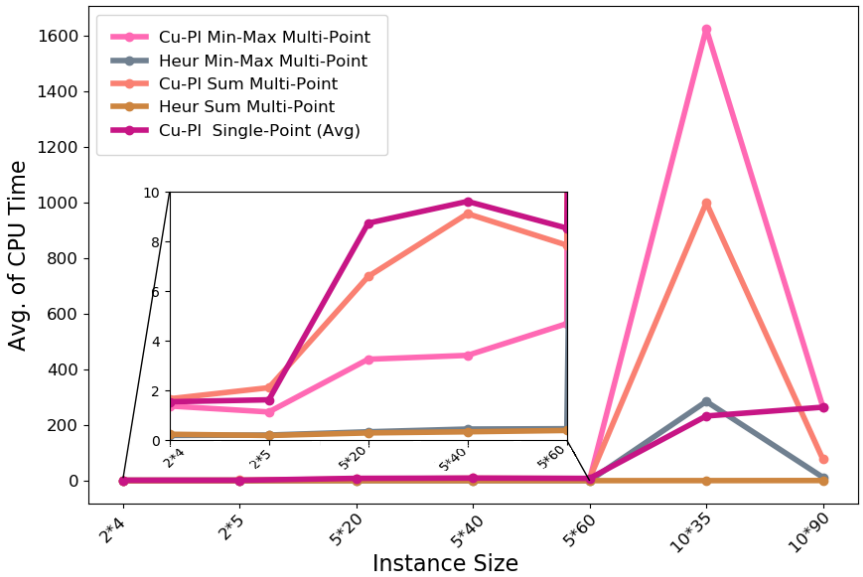
Data preprocessing (Python)

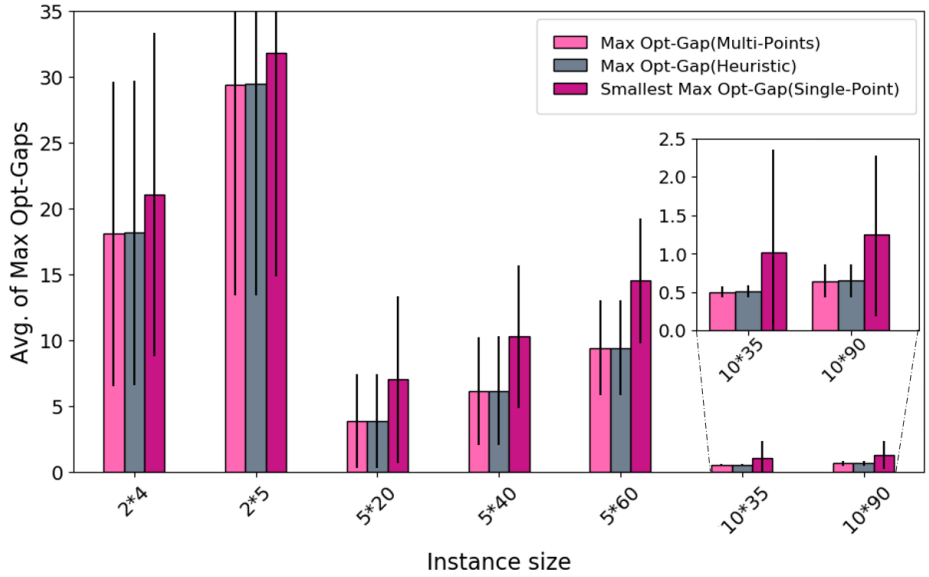
- Data **cleaning**.
- Identifying and handling the **missing values**.
- **Feature reduction and selection**.
- **Encoding** the categorical data.
- **Splitting data**(train,test 80/20)



Training to Find Matrices

- Cutting-Plane Algorithm vs LP-Relaxation Heuristic.





Testing to do Predictions

- **Accuracy** tells if the recommender system is able to **predict** those items that **have already been rated**.

Size of Instances	Avg % of Finding Actual Items			
Number of Traning Users	Cu-Pl Sum	Heur Sum	Cu-Pl Min-Max	Heur Min-Max
5 users	71 %	78 %	69 %	70 %
10 Users	81 %	87 %	80 %	82 %
15 Users	86 %	83 %	85 %	80 %
20 Users	81 %	73 %	81 %	76 %
25 Users	87 %	86 %	87 %	76 %

Illustrating Accuracy (C++ using Concert Technology (CPlex).)

Contents

Problem Statement

Methodology

Data & Experiments

Conclusion

Results

- **Cutting plane** and **heuristic** with norm 1 performs better than norm infinity.
- The **CPU time** decrease using **heuristics**.
- The **average of optimality gaps** differences between predicted and actual rated items are pretty small.
- IO for RS aims to optimize the **accuracy**(more than 70 %).

- IO is a **data-driven** technique for prediction.
- It **optimizes decision-making** using historical data.

Inverse Optimization	Machine learning
Cutting-plane algorithm, Heuristics	Collaborative filtering algorithms
Users are optimizing decisions	Users are satisfied with decisions
Finding the matrix needs small data	Finding the matrix needs large datasets
There are no prior item ratings	Some items have already been rated

Similarities:

- ✓ They are data driven techniques for predictive analytics.
- ✓ They learn from historical data and similarities with past users
- ✓ They can be applied for a real time case.

Application Directions & Challenges

- Considering **dynamic behavior** of users.
- Using feature selections.
- Adding **time, location** constraints.
- **Clustering** users in data sets.
- **Comparing** IO with **machine learning** techniques.

Challenges for big-data and real-case

To make it faster for a robust request we can:

- Cluster data.
- Use data in the same cluster with the new customer.
- train the model to learn matrix.
- Recommend based on the optimal items to the customer.


QUESTIONS?
COMMENTS?
CONCERNS?