Tutorial: Multi-Objective Optimization and Recommendations

Yong Zheng, Illinois Institute of Technology, USA
David (Xuejun) Wang, Morningstar, Inc., USA

at the IEEE International Conference on
Data Mining (ICDM) 2022

Tutorial Schedule

- Part 1: Multi-Objective Optimization (MOO)
  - Presenter: Dr. David (Xuejun) Wang
  - Time: 10:30 AM – 12:00 PM

------------- Lunch: 12:00 – 1:00 PM -------------

- Part 2: Recommender Systems with MOO
  - Presenter: Dr. Yong Zheng
  - Time: 1:00 PM – 3:00 PM
- Website: https://moorecsys.github.io/

Part 2: Recommender Systems with Multi-Objective Optimization

Yong Zheng
Illinois Institute of Technology, USA

Time: 1:00 – 3:00 PM

https://moorecsys.github.io/

Agenda

01

Intro

Intro to RecSys

02

Why MOO

Contexts in which
we need MOO in RecSys

03

Case Studies

Examples of using
MOO in RecSys

04

Summary

Summary, Guideline,
Challenges & QA

Agenda

01

Intro

Intro to RecSys

02

Why MOO

Contexts in which
we need MOO in RecSys

03

Case Studies

Examples of using
MOO in RecSys

04

Summary

Summary, Guideline,
Challenges & QA

Recommender Systems (RecSys)

- Item recommendations tailored to user preferences

How it works

- User Preferences on the items

Ratings

Binary Feedback

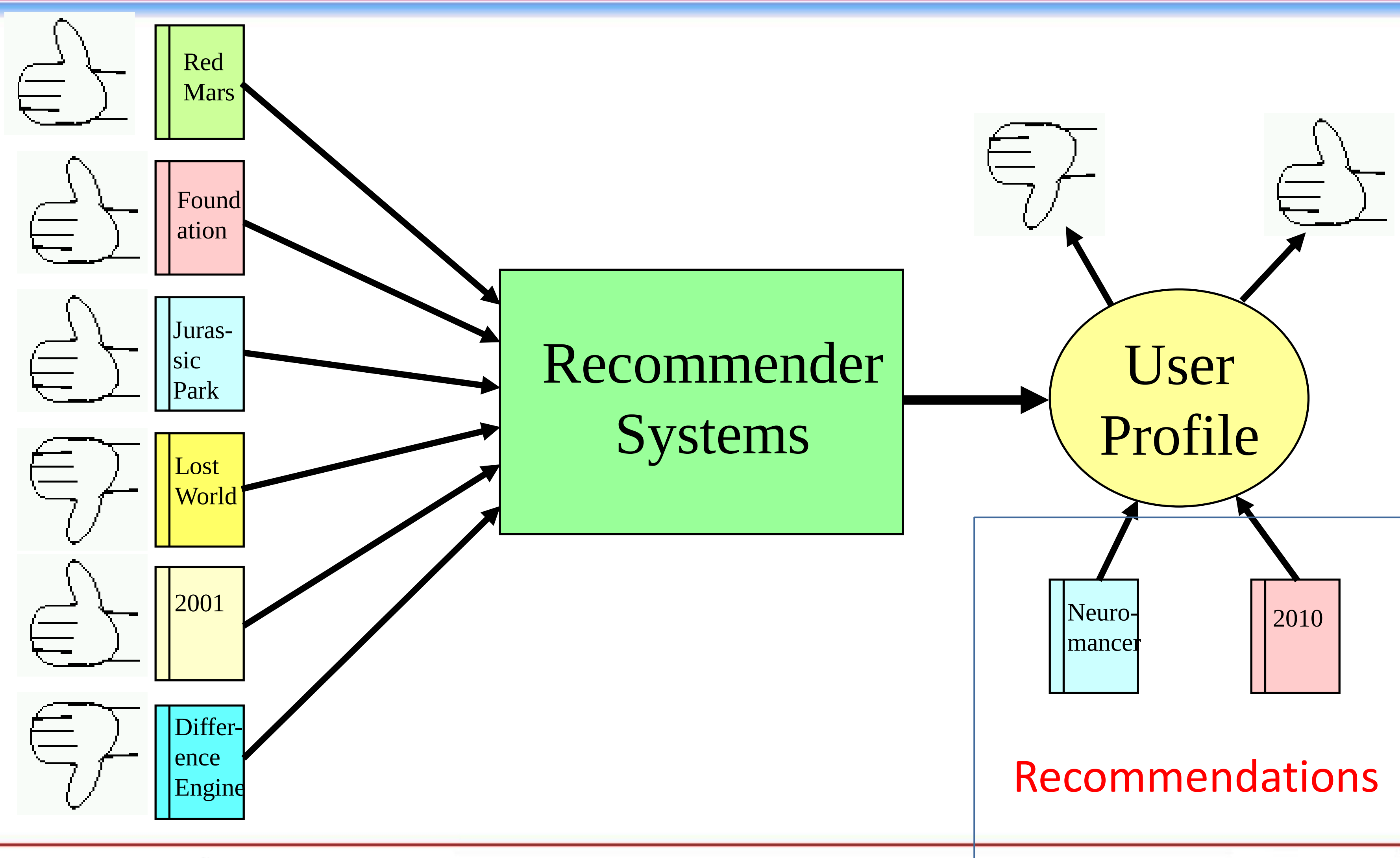
Reviews

Behaviors

Explicit

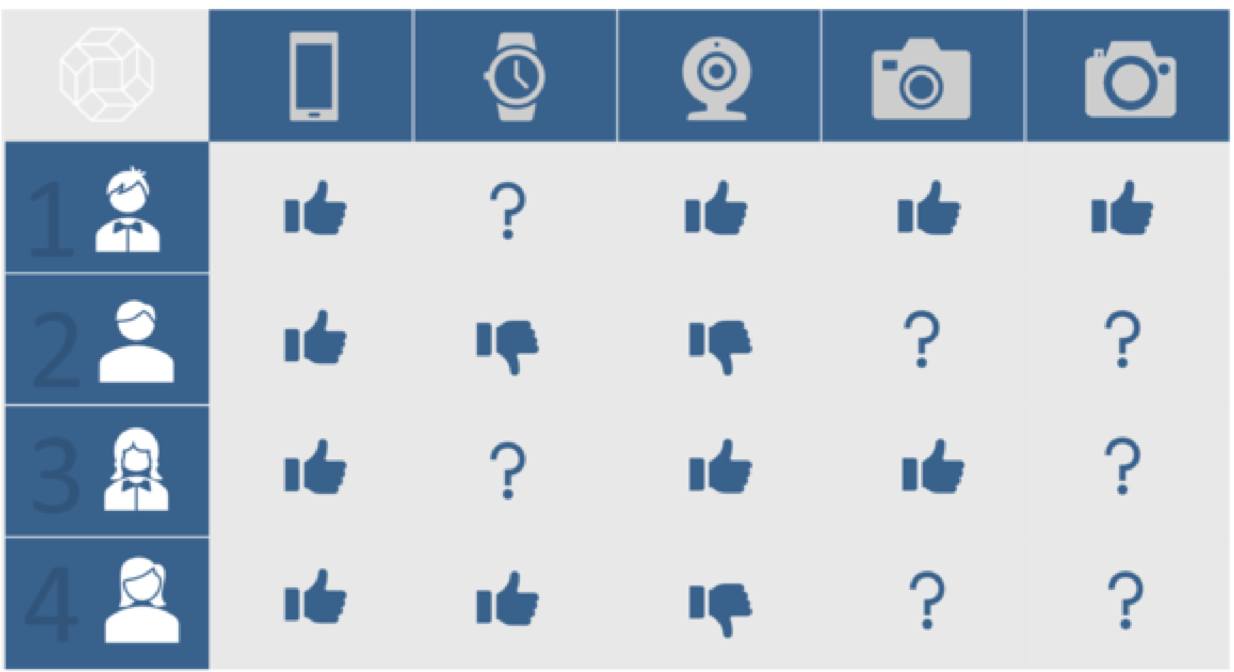
Implicit

How it works


Red Mars
Foundation
Jurassic Park
Lost World
2001
Difference Engine
Recommender Systems
User Profile
Neuromancer
2010
Recommendations

Traditional Recommender Systems

- User demographic information + Item features
- Users’ preferences on the items



Different Types of Recommender Systems

Context-Aware RecSys: Incorporate context info (time, location, etc) into RecSys

Group RecSys: Recommend items to a group of users, e.g., group dinning

Multi-Stakeholder RecSys: Produce recommendations by considering multiple stakeholders, e.g., buyers and sellers on eBay

Multi-Task RecSys: Build joint learning model by considering multiple tasks, e.g., RecSys + opinion texts

Recommendation Algorithms

- Memory-Based Approaches
  - User-Based & Item-Based Collaborative Filtering
- Model-Based Approaches
  - Optimize objectives by machine learning
  - E.g., matrix factorization, deep learning models
- Content-Based Models
- Hybrid Models

Agenda

01

Intro

Intro to RecSys

02

Why MOO

Contexts in which
we need MOO in RecSys

03

Case Studies

Examples of using
MOO in RecSys

04

Summary

Summary, Guideline,
Challenges & QA

Single-Objective Recommender Systems

- Model-Based RecSys
  - It is usually a process of single-objective optimization
  - Example: Minimizing errors in *rating predictions*

$$\min_{q^*,p^*} \sum_{(u,i)\in\kappa} \underbrace{(r_{ui} - q_i^T p_u)^2}_{\text{Sum of squared errors}} + \underbrace{\lambda(\| q_i \|^2 + \| p_u \|^2)}_{\text{regularization term}}$$

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8), 30-37.

Single-Objective Recommender Systems

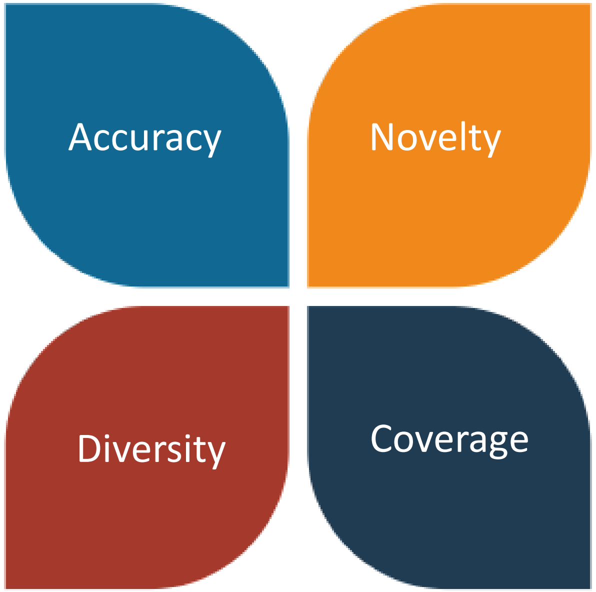
- Model-Based RecSys
  - It is usually a process of single-objective optimization
  - Example: Maximizing ranking in *learning-to-rank*

$$\min \sum_{(u,i,j)\in D_S} \underbrace{-\ln \sigma(p_u^T q_i - p_u^T q_j)}_{\text{Pairwise ranking loss}} + \underbrace{\lambda(||p_u||^2 + ||q_i||^2)}_{\text{regularization term}}$$

Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2012). BPR: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618.

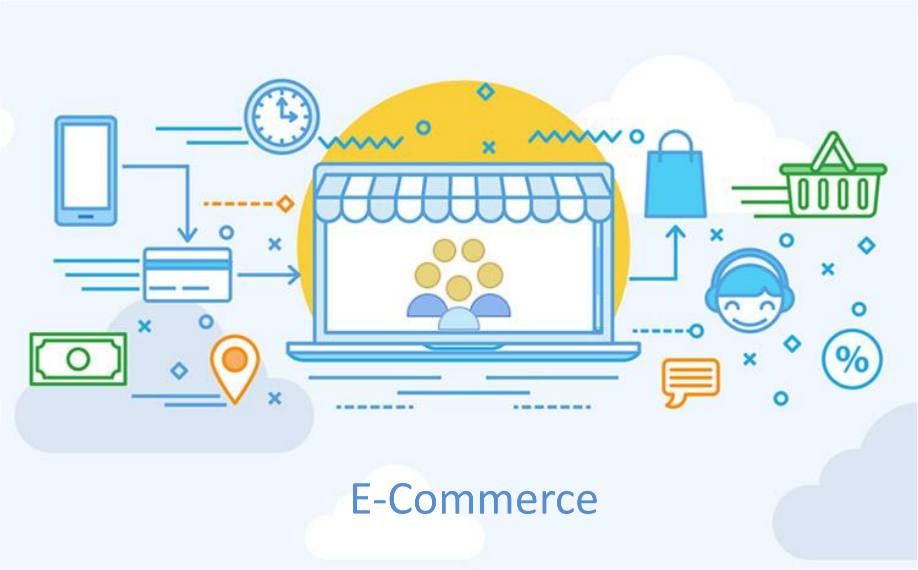
Why MOO in RecSys

- There is an emerging demand in MOO
  - Traditional RecSys
    - Example: RecSys balancing multiple metrics, e.g., news



Why MOO in RecSys

- There is an emerging demand in MOO
  - New Types of RecSys
    - Example: Multi-stakeholder RecSys, e.g., marketplace



RecSys with MOO

- Motivations of Using MOO in RecSys
  1) Finding a balance among multiple objectives
     - Considering multiple RecSys metrics
     - Considering objectives from multiple stakeholders
     - Looking for improvements on multiple tasks
     - ….
  2) Improve RecSys by considering multiple objectives
     - Improving Group RecSys by considering diff objectives
     - Improving RecSys by considering objs in pre-processing
     - …

RecSys with MOO

- How to use MOO/its Knowledge in RecSys
  1) Recommendation Task as a MOO Process
     - A MOO is involved in recommendation process
     - MOO techniques introduced by David can be reused to serve in recommender systems
  2) Enhanced RecSys with Dominance Relations
     - There are is multi-objective optimization process
     - Knowledge or skills (e.g., dominance relations) from MOO can be reused to improve recommendations

Agenda

01

Intro

Intro to RecSys

02

Why MOO

Contexts in which
we need MOO in RecSys

03

Case Studies

Examples of using
MOO in RecSys

04

Summary

Summary, Guideline,
Challenges & QA

RecSys with MOO

- How to use MOO/its Knowledge in RecSys
  1) Recommendation Task as a MOO Process
     - A MOO is involved in recommendation process
     - MOO techniques introduced by David can be reused to serve in recommender systems
  2) Enhanced RecSys with Dominance Relations
     - There are is multi-objective optimization process
     - Knowledge or skills (e.g., dominance relations) from MOO can be reused to improve recommendations

RecSys with MOO

- How to use MOO/its Knowledge in RecSys
  1) Recommendation Task as a MOO Process
     - A MOO is involved in recommendation process
     - MOO techniques introduced by David can be reused to serve in recommender systems
  2) Enhanced RecSys with Dominance Relations
     - There is no multi-objective optimization process
     - Knowledge or skills (e.g., dominance relations) from MOO can be reused to improve recommendations

RecSys with MOO

- Contexts in which we need MOO in RecSys

RecSys with MOO

- Rcap: MOO methods
  - Scalarization
    - Weighting Methods
    - ϵ-Constraint Method
    - Normal Constraint (NC)
    - …
  - Multi-objective evolutionary algorithms (MOEA)
    - GA-based MOEA
    - PSO-based MOEA
    - ….

RecSys with MOO

- Contexts in which we need MOO in RecSys

- RecSys Balancing Multiple Metrics
- User-Based Collaborative Filtering
- Preprocessing for RecSys
- Group RecSys
- Multi-Stakeholder RecSys
- Multi-Task RecSys

RecSys with MOO

- For each category

Why	Objectives	Goals	Case Studies
Why we need MOO in these applications	How the objectives are defined	The goals of these RecSys	Examples of utilizing scalarization & MOEA

RecSys with MOO

- Contexts in which we need MOO in RecSys

RecSys balancing multiple metrics

- Why we need MOO in this context
  - Relevance or accuracy is not the only focus
    - For example, news and music recommendations
      - Boring if always recommending the same types of items
      - Diversity: try something different
      - Novelty: try something never experienced before
    - For example, item recommendations in e-commerce
      - Co-sales
      - Bundle sales

RecSys balancing multiple metrics

- Objective definitions in this context
  - Optimize more metrics in addition to accuracy

Accuracy

Relevance of recommendations
e.g., precision, recall, NDCG, etc

Novelty

Unknown to the user,
but potentially interested in

Diversity

Recommend something different,
e.g., different item categories

Coverage

User coverage & item coverage

Kaminskas, M., & Bridge, D. (2016). Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems. ACM TIIS, 7(1), 1-42.

RecSys balancing multiple metrics

- Goals
  - Improve other metrics at no loss or acceptable/limited loss on accuracy
  - Challenges
    - No clear rules to define the “acceptable/limited” loss, without more work in user studies

RecSys balancing multiple metrics

- Case Studies
  - Using MOEA in MOO process

    Ribeiro, M. T., Lacerda, A., Veloso, A., & Ziviani, N. (2012). Pareto-efficient hybridization for multi-objective recommender systems. In ACM RecSys 2012.

    Decision variables: weights in aggregations

  - Using MOEA in MOO process

    Chai, Z., Li, Y., & Zhu, S. (2021). P-MOIA-RS: a multi-objective optimization and decision-making algorithm for recommendation systems. Journal of Ambient Intelligence and Humanized Computing, 12, 443-454.

    Decision variables: recommended items

    Note: most of the research in this category utilized MOEA as the MOO method

RecSys balancing multiple metrics

- Case Study 1: Hybrid Recommender

  Ribeiro, M. T., Lacerda, A., Veloso, A., & Ziviani, N. (2012). Pareto-efficient hybridization for multi-objective recommender systems. In ACM RecSys 2012.

  – Application: balancing accuracy, novelty, diversity

  – Recommendation Framework

Step 1
Build multiple recommendation models to predict rating for a user on an item

Step 2
Fuse predictions by weighted sum

$$\hat{r}_{ui} = w_1 \times \hat{r}_{ui,1} + w_2 \times \hat{r}_{ui,2} + \ldots + w_n \times \hat{r}_{ui,n}$$

Step 3
Produce recommendation list by the aggregated rating

Step 4
Learn the optimal weights (decision variables) by considering accuracy, novelty, diversity

RecSys balancing multiple metrics

- Case Study 1: Hybrid Recommender

  Ribeiro, M. T., Lacerda, A., Veloso, A., & Ziviani, N. (2012). Pareto-efficient hybridization for multi-objective recommender systems. In ACM RecSys 2012.

  – MOEA as the MOO Method
    - Consider accuracy, diversity, novelty as objectives
    - Use Strength Pareto Evolutionary Algorithm as MOEA optimizer
      – Encoding/Decision variables: the weights in the hybrid model
      – Output: a Pareto optimal set
    - Select the best single solution from Pareto set
      – Use a weighted sum on the three objectives
      – Try different set of weights (Q_j) manually

$$\underset{i \in P}{arg\ max} \sum_{j=1}^{|O|} Q_j O_{ij}$$

  – Results: balancing multiple metrics

RecSys balancing multiple metrics

- Case Study 2: Learn recommendations by MOEA

Chai, Z., Li, Y., & Zhu, S. (2021). P-MOIA-RS: a multi-objective optimization and decision-making algorithm for recommendation systems. Journal of Ambient Intelligence and Humanized Computing, 12, 443-454.

– Recommendation Model

Step 1
Use item-based collaborative filtering (IBCF) to produce top-K list

Step 2
Utilize MO- Immune Algorithm to produce top-N list (decision variables) from top-K items (N<<K)

Step 3
Produce Pareto Set by considering accuracy & diversity

Step 4
Select the best solution by PROMETHEE

– Notes: Using IBCF to produce top-K list is an optional step

RecSys with MOO

- Summary
  - Most of the research in this category utilized MOEA to balance accuracy, diversity, novelty, etc
  - The method in case study #2 is more general –> learning the recommendation list directly
  - The method in case study #1 is a special case to be applied on hybrid recommender systems

RecSys with MOO

- Contexts in which we need MOO in RecSys

User-based Collaborative Filtering (UBCF)

- Why we need MOO in this context
  - Assumption: the issue of accuracy and diversity in the recommendations by UBCF is relevant with the *neighborhood selection* in UBCF
  - Considering both user-user similarities and diversity may deliver more diverse recommendations

User-based Collaborative Filtering (UBCF)

- Objective definitions
  - Similarity (u, N_u), while N_u is set of user neighbors
  - Diversity (N_u) = intra-group diversity of neighbors
- Goals
  - Improve or balance accuracy and diversity

User-based Collaborative Filtering

- Case Study: Using scalarization as the MOO method

  Karabadji, N. E. I., Beldjoudi, et al. (2018). Improving memory-based user collaborative filtering with evolutionary multi-objective optimization. Expert Systems with Applications, 98, 153-165.

  – Scalarization method + optimization by using GA

Step 1

Setup single-objective task

$= \alpha \cdot \mathit{Similarity} + (1-\alpha) \cdot \mathit{Diversity}$

Step 2

Use GA as the optimizer
Encoding: a set of neighbors for each user (decision variables)

Step 3

User UBCF to utilize learned neighbors in RecSys

Step 4

Tune up weights and evaluate multiple metrics

  – Notes
    - MOO is used for neighbor selection only
    - GA is used as a single-objective optimizer

User-based Collaborative Filtering (UBCF)

- Summary
  - These work aimed to improve accuracy and diversity but were designed for UBCF specifically
  - Drawback: the assumption that the diversity of the neighbors resulting in diversity in recommendations may not be always true

RecSys with MOO

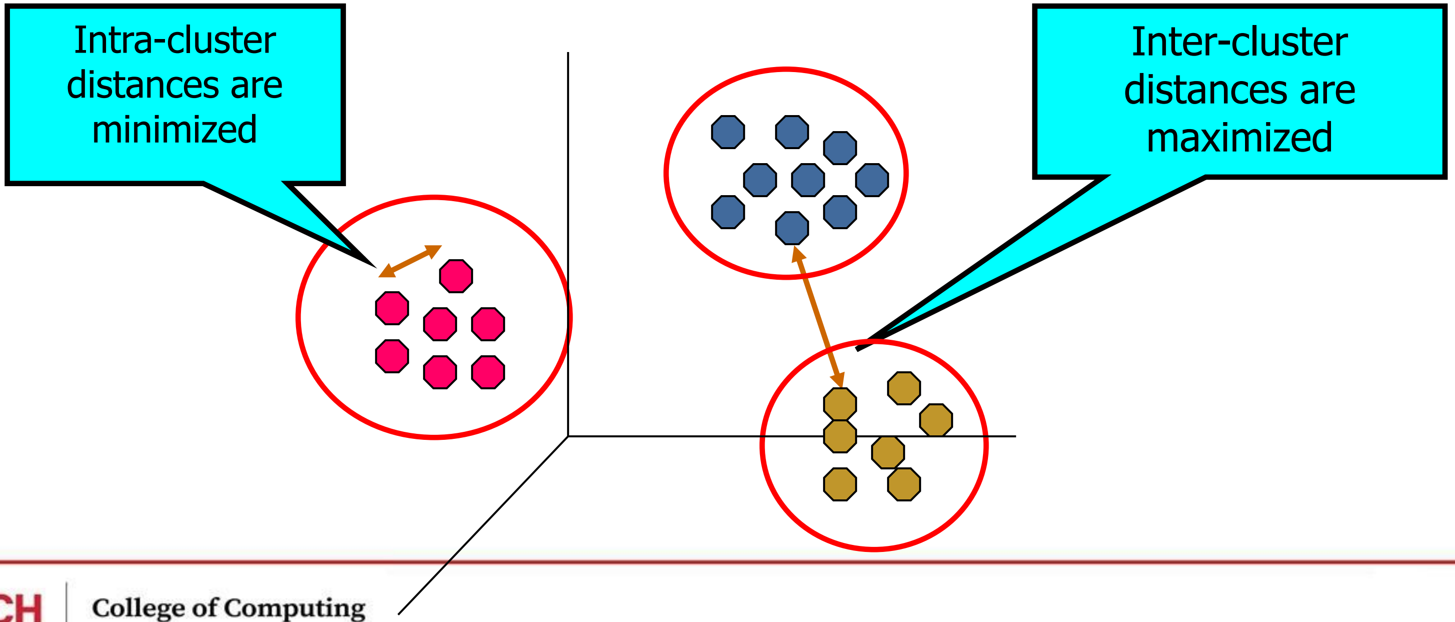
- Contexts in which we need MOO in RecSys

Preprocessing for RecSys

- Why we need MOO in this context
  - Unsupervised learning may be used as the intermediate process in some recommendation models
    - Clustering to be used to create user or item clusters
    - Association rules to be produced in order to assist rule-based recommendation models
  - Using MOO to produce better outputs which can assist RecSys

Preprocessing for RecSys

- Objective definitions
  - Clustering, e.g.,
    - *Intra-cluster distance* to get a dense cluster
    - *Inter-cluster distance* to separate with other clusters



Preprocessing for RecSys

- Objective definitions
  - Association Rules, e.g.,
    - *Support* to produce frequent item sets
    - *Confidence* to generate useful rules

$$Rule: X \Rightarrow Y$$

$$Supprt = \frac{Frequency(X,Y)}{N}$$

$$Confidence = \frac{Frequency(X,Y)}{Frequency(X)}$$

$$Lift = \frac{Support}{Support(X) \times Support(Y)}$$

- Goals
  - better outputs to assist RecSys

Preprocessing for RecSys

- Case Study: Using scalarization in association rules
  - Application: produce rules like $(T_1, T_2) \rightarrow T_3$. If a user likes T_1 and T_2, it infers that T_3 is a good recommendation candidate

Step 1
Setup single-objective task
$w_1 \times sup(A \rightarrow B) + w_2 \times conf(A \rightarrow B)$

Step 2
Use GA as the single-objective optimizer to learn the rules (decision variables)

Step 3
Find a Pareto set by grid search of w_1 in [0,1]

Step 4
Select the optimal solution by F_1 in RecSys

  - Results: better than using MOPSO (a MOEA method)

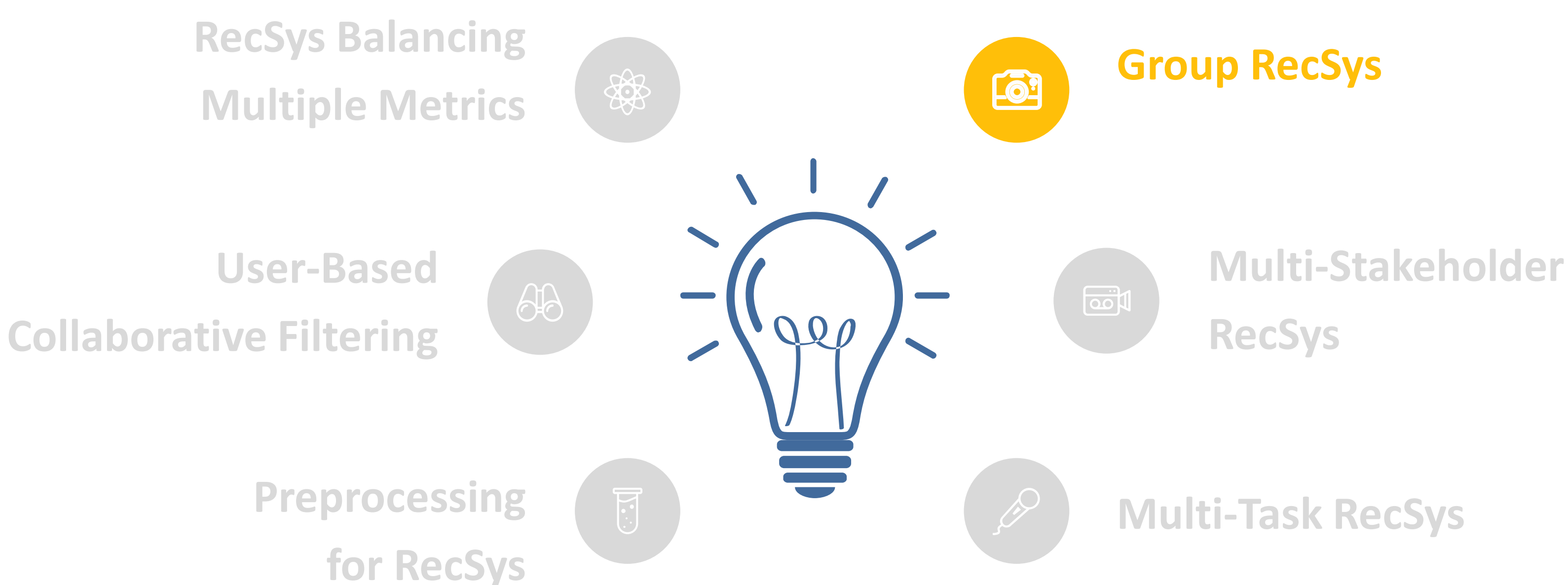
Neysiani, B. S., Soltani, N., Mofidi, R., & Nadimi-Shahraki, M. H. (2019). Improve performance of association rule-based collaborative filtering recommendation systems using genetic algorithm. Int. J. Inf. Technol. Comput. Sci, 11(2), 48-55.

Preprocessing for RecSys

- Summary
  - MOO for preprocessing or unsupervised learning (clustering, rule mining, etc.) were well-developed
  - The number of these applications in RecSys is limited

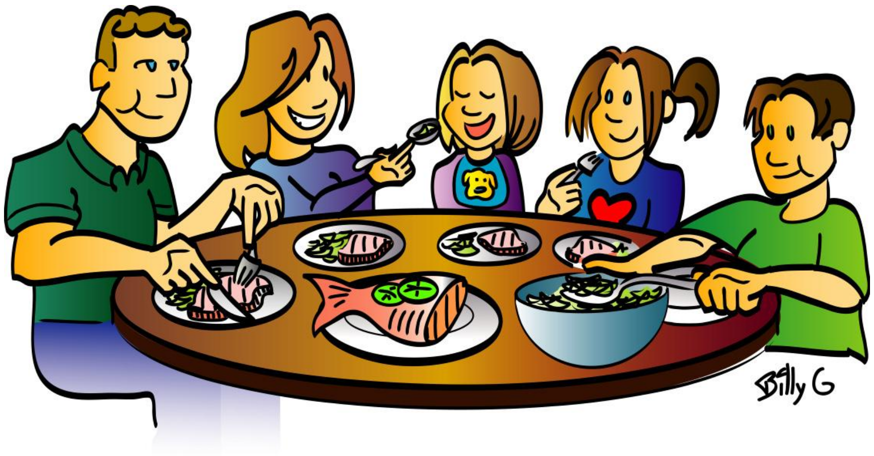
RecSys with MOO

- Contexts in which we need MOO in RecSys



Group RecSys

- Why we need MOO in this context
  - Group RecSys: produce item recommendations to a group of users, such as group dinner or travel
  - Key factors
    - Individual tastes
    - Group satisfaction



Group RecSys

- Objective definitions
  - Individual satisfaction
  - Group fairness/satisfaction
- Goals
  - Produce better group recommendations

Group RecSys

- Case Study: Using scalarization

Xiao, L., Min, Z., Yongfeng, Z., Zhaoquan, G., Yiqun, L., & Shaoping, M. (2017, August). Fairness-aware group recommendation with pareto-efficiency. In ACM RecSys, 2017

individual satisfaction $U(u, I) = \frac{1}{K \times rel_{max}} \sum_{i \in I} rel(u, i)$

I = a set of K recommended items
u = a member in the group g

Objective 1: social welfare/ average individual satisfaction

$$SW(g, I) = \frac{1}{|g|} \sum_{u \in g} U(u, I), \forall g, I$$

Objective 2: group fairness

$$Least\ Misery: \ F_{LM}(g, I) = \min\{U(u, I), \forall u \in g\}$$

$$Min - Max\ Ratio: \ F_M(g, I) = \frac{\min\{U(u, I), \forall u \in g\}}{\max\{U(u, I), \forall u \in g\}}$$

Group RecSys

- Case Study: Using scalarization

  Xiao, L., Min, Z., Yongfeng, Z., Zhaoquan, G., Yiqun, L., & Shaoping, M. (2017, August). Fairness-aware group recommendation with pareto-efficiency. In ACM RecSys, 2017

  – MOO methods
    - Scalarization $\lambda \cdot SW(g, I) + (1 - \lambda) \cdot F(g, I)$
    - Optimization: greedy search/integer programming

  – Results
    - By considering group fairness, it is able to improve group recommendations in terms of F_1 and NDCG

Group RecSys

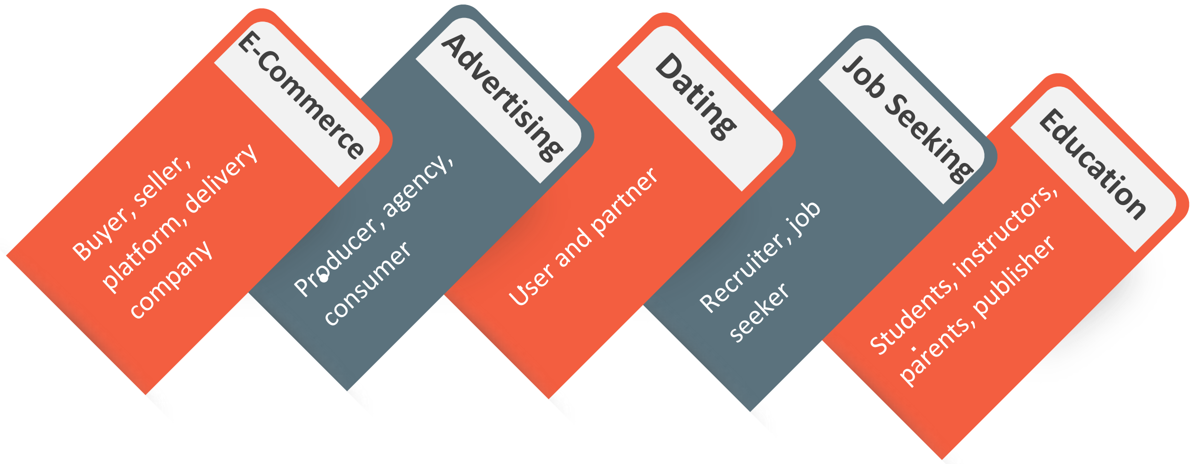
- Summary
  - Using MOO in Group RecSys is promising, but the number of research work is limited in the current stage
  - The goal is to improve group RecSys, by taking multiple objectives into considerations

RecSys with MOO

- Contexts in which we need MOO in RecSys

Multi-Stakeholder RecSys

- Why we need MOO in this context
  - The end user is not the only stakeholder



  - RecSys should be built by considering the item utility from the perspective of different stakeholders

Multi-Stakeholder RecSys

- Objective definitions
  - It varies from domains to domains
  - For each stakeholder, there’s at least one objective
    - E-Commerce or Marketplace
      - Buyer: user preferences on items, budget
      - Seller: profits
      - Platform: commission fees
      - Delivery company: costs and profits
    - Job seeking
      - Job seeker: user preferences
      - Recruiter: talent requirements

Multi-Stakeholder RecSys

- Goals
  - Deliver item recommendations by balancing the needs of multiple stakeholders
  - With acceptable loss on the consumer side
  - Challenges
    - Which stakeholders should be considered
    - How to define and achieve the “balance”
    - No clear rules to define the acceptable loss

Multi-Stakeholder RecSys

- Case Studies
  - Using scalarization as the MOO method

    Lin, X., Chen, H., Pei, C., et al. (2019). A pareto-efficient algorithm for multiple objective optimization in e-commerce recommendation. In ACM RecSys, 2019.

  - Using MOEA as the MOO method

    Zheng, Y., Ghane, N., & Sabouri, M. (2019). Personalized educational learning with multi-stakeholder optimizations. In Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization (pp. 283-289).

Multi-Stakeholder RecSys

- Case Study 1: Using scalarization in E-Commerce
  - Objectives
    - CTR (Click Through Rate)
    - GMV (Gross Merchandise Volume)

Lin, X., Chen, H., Pei, C., et al. (2019). A pareto-efficient algorithm for multiple objective optimization in e-commerce recommendation. In ACM RecSys, 2019.

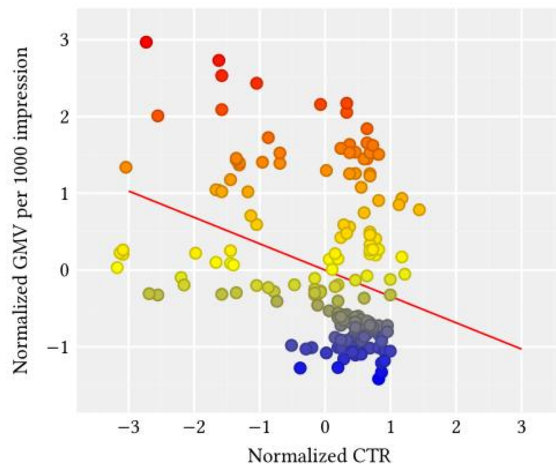


Figure 1: The trade-off between CTR and GMV. The Pearson Correlation Coefficient is -0.343086, with $p < 0.01$.

Multi-Stakeholder RecSys

- Case Study 1: Using scalarization in E-Commerce
  - MOO Method
    - Define a loss function for each objective

      CTR $\mathcal{L}_{CTR}(\theta, x, y, z) = -\frac{1}{N}\sum_{j=1}^{N} log(P(y_j|\theta, x_j))$, i.e., point-wise learning-to-rank

      GMV $\mathcal{L}_{GMV}(\theta, x, y, z) = -\frac{1}{N}\sum_{j=1}^{N} h(price_j) \cdot log(P(z_j = 1|\theta, x_j))$

      x: impression, y: clicks, z: purchases

    - Use weighted sum as the scalarization
      Joint Loss = $\omega \cdot L_{CTR} + (1 - \omega) \cdot L_{GMV}$

Multi-Stakeholder RecSys

- Case Study 1: Using scalarization in E-Commerce
  - MOO Method
    - Use weighted sum as the scalarization
      Joint Loss = $\omega \cdot L_{CTR} + (1 - \omega) \cdot L_{GMV}$
    - Try different weights to get Pareto Set
    - Select a single best solution by using Least Misery strategy, i.e., minimizing the highest loss function of the objectives

$$\min \max\{\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_K\}$$

Multi-Stakeholder RecSys

- Case Study 1: Using scalarization in E-Commerce
  - Results
    - Present improved NDCG and other metrics (e.g., GMV,) in both offline and online experiments

Approaches	CTR	IPV	PAY	GMV
CXR-RL	13.68	20.60	-1.027	-3.197
PO-EA	4.442	8.957	3.399	-3.038
PE-LTR	**13.80***	**23.76***	**20.09***	**3.623***

Three-days online experiments

Table presents improvement ratio over a same baseline

CTR (Click Through Rate)
IPV (Individual Page View)
PAY (number of payments)
GMV (Gross Merchandise Volume)

Multi-Stakeholder RecSys

- Case Study 2: Using MOEA in Education
  - Application: recommending Kaggle data sets to students for their data science projects
  - Data: both instructors and students had multi-criteria ratings on the data, and their expectations in shape of the multi-criteria rating too

Table 1: Example of The Educational Data

User	Item	Overall Rating	App	Data	Ease
10	41	4	4	4	4
10	60	2	2	2	2
12	21	4	4	5	4
...	...	...	...	...	...

Table 2: User Expectation Data

User	App	Data	Ease
10	5	4	3
12	4	4	4
...	...	...	...

Multi-Stakeholder RecSys

- Case Study 2: Using MOEA in Education
  - Conflicting Interests
    - Instructors: do not expect students to select projects that are too easy, but also give them the chance to select their preferred ones
    - Students: someone prefer challenging ones; someone like easy ones
  - Objectives
    - Maximizing item utility from perspective of students
    - Maximizing item utility from perspective of instructors
    - Maintaining limited loss in recommendations, e.g., NDCG

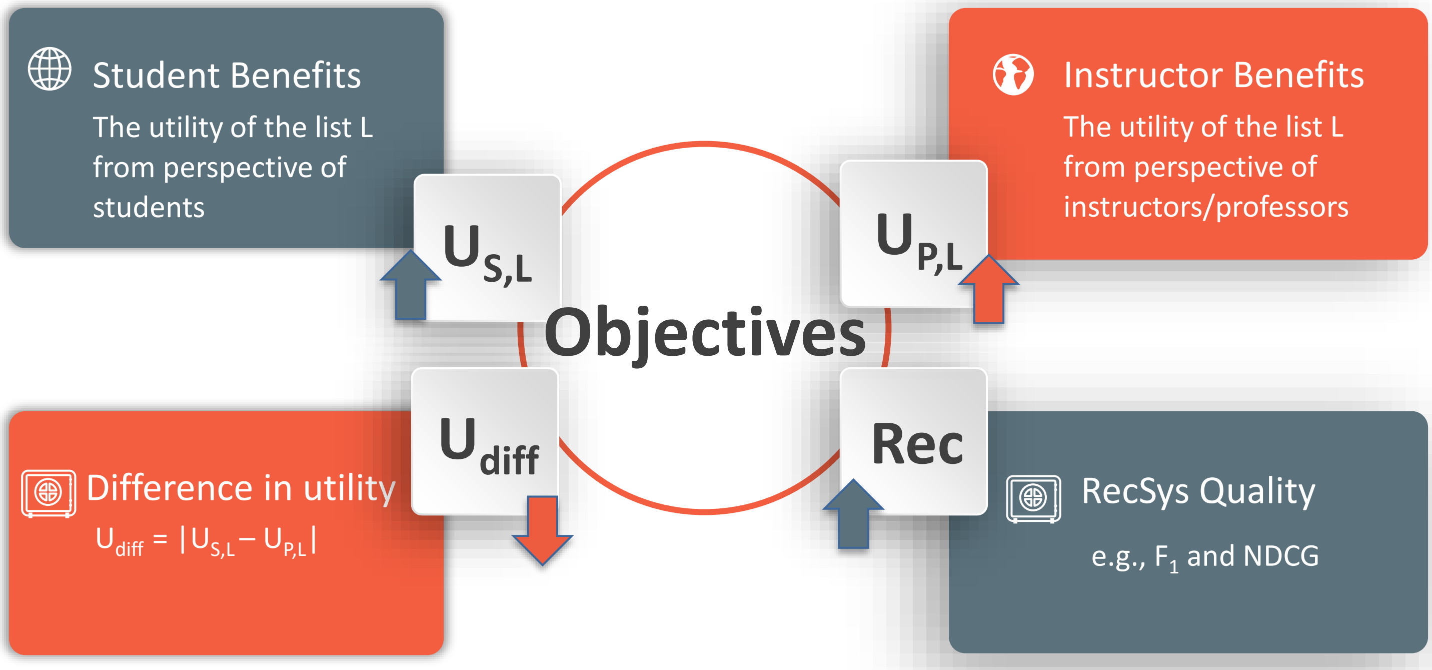
Zheng, Y., Ghane, N., & Sabouri, M. (2019). Personalized educational learning with multi-stakeholder optimizations. In Adjunct Publication of the 27th Conference on UMAP (pp. 283-289).

Multi-Stakeholder RecSys

- Case Study 2: Using MOEA in Education
  - A utility-based recommendation model
    - Utility can be denoted by similarity between multi-criteria rating vector (R) and expectation vectors (E)
    - Student, $U_{s,t}$ = similarity ($E_{s,}$, $R_{s,t}$)
    - Instructor, $U_{p,t}$ = similarity (E_p , $R_{p,t}$)
    - Ranking score to sort items = $\alpha \times U_{s,t} + (1 - \alpha) \times U_{p,t}$

Utility-Based Multiple Stakeholder Recommendation

- Case Study 2: Using MOEA in Education



Multi-Stakeholder RecSys

- Case Study 2: Using MOEA in Education
  - MOO methods
    - Using MOEA as the multi-objective optimizer
      - Open-Source MOEA, http://moeaframework.org
      - Demo, https://github.com/irecsys/Tutorial_MSRS
    - MOEA will produce a Pareto set
    - Select the single best solution based on TOPSIS
      - Calculate the maximal objectives by using single-objective recommendation model, e.g. maximizing recommendation qualities by considering students/instructors only
      - Then calculate the average loss of the objectives
      - The solution with minimal loss is the best one

Multi-Stakeholder RecSys

- Case Study 2: Using MOEA in Education
  - Results
    - Balancing the needs of instructors and students at a small loss at recommendations (NDCG & F_1)

	$U_{S,L}$	$U_{P,L}$	F_1	NDCG	Loss
UBRec	0.181	0.134	0.085	0.126	0.180
$Rank_p$	0.072	0.298	0.027	0.039	0.425
MSRS	0.199	0.251	0.074	0.107	0.063

    - UBRec: the best model considering students only
      $Rank_p$: the best model considering instructors only

Multi-Stakeholder RecSys

- Summary
  - The nature of multi-stakeholder RecSys is involved with a process of multi-objective optimization
  - Multi-stakeholder and multi-task RecSys are two major applications of using MOO in RecSys

RecSys with MOO

- Contexts in which we need MOO in RecSys

Multi-Task RecSys

- Why we need MOO in this context
  - Multi-task RecSys refer to the recommender systems which optimize multiple tasks by a joint learning process
  - Joint learning is not novel, but multi-task RecSys usually share some common representations
    - Latent factors
    - Feature spaces
    - Neural network layers
    - …..

Multi-Task RecSys

- Objective Definitions
  - It varies from tasks to tasks

Multiple Recommendation Tasks

e.g., rating prediction task and ranking task

Multiple Feedbacks or User Behaviors

e.g., Click-through, View-through, # of purchases, etc

RecSys Task + Non-RecSys Task

Text Generations: reviews, tips, opinions, explanations

Trust or social predictions

Others: classifications, regressions, etc

Multi-Task RecSys

- Goals
  - Improve tasks by a joint learning process
  - Assumption: the improvement is dependent with the correlation of the tasks and the power of the shared representations

Multi-Task RecSys

- Case Studies
  - Using Scalarization in Multiple RecSys tasks

    Hadash, G., Shalom, O. S., & Osadchy, R. (2018). Rank and rate: multi-task learning for recommender systems. In ACM RecSys 2018.

  - Using Scalarization in RecSys + Text tasks

    Li, P., Wang, Z., Ren, Z., Bing, L., & Lam, W. (2017). Neural rating regression with abstractive tips generation for recommendation. In ACM SIGIR 2017.

Multi-Task RecSys

- Case Study 1: Scalarization in RecSys Tasks

  Hadash, G., Shalom, O. S., & Osadchy, R. (2018). Rank and rate: multi-task learning for recommender systems. In ACM RecSys 2018.

  – Application: build a joint learning model which optimize the rating prediction and ranking tasks

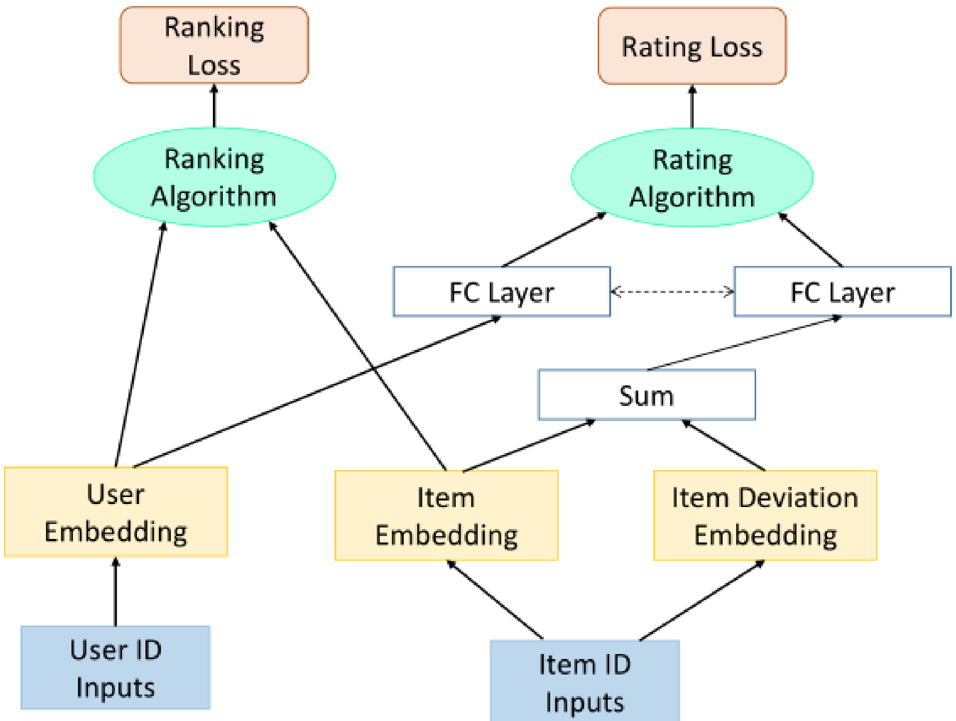
  – Why?

    - The results in the rating prediction task are not always consistent with the results in ranking

Multi-Task RecSys

- Case Study 1: Scalarization in RecSys Tasks

Hadash, G., Shalom, O. S., & Osadchy, R. (2018). Rank and rate: multi-task learning for recommender systems. In ACM RecSys 2018.



Recommendation Models

- A multi-task framework
- Share user & item embeddings
- A joint learning process

  Loss = weighted sum of L_R & L_P

$$O = \min_{U,I} \alpha \cdot L_R(\mathcal{R}, D; U, I) + (1-\alpha) \cdot L_P(\mathcal{P}, D; U, I) + \lambda(\|U\|^2 + \|I\|^2).$$

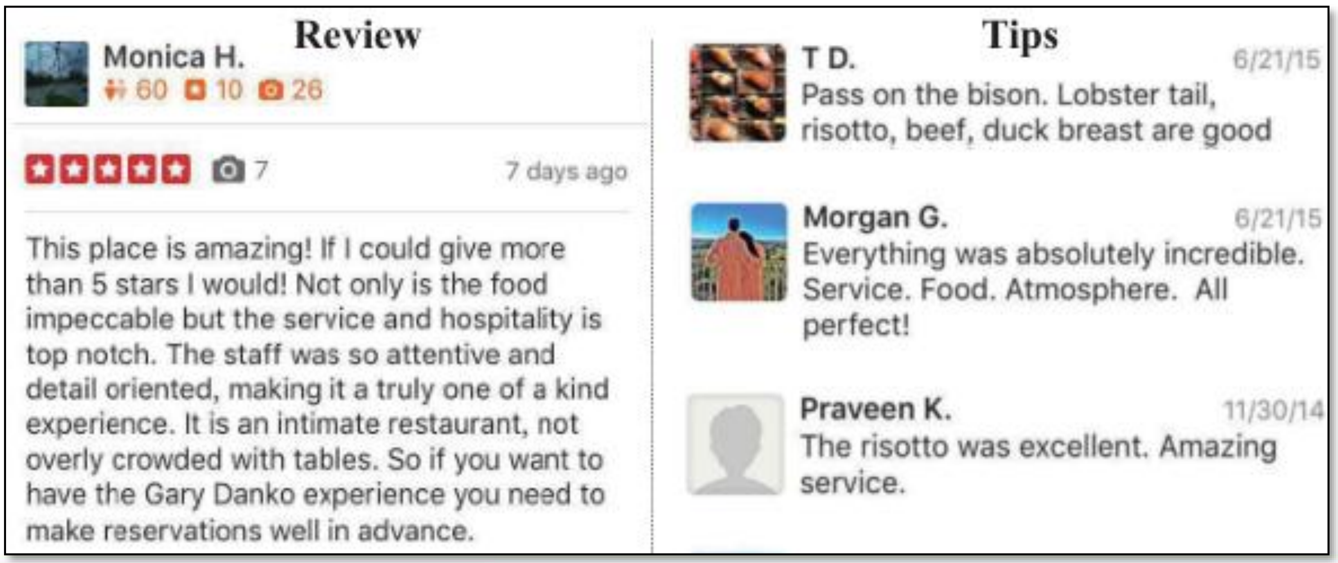
- Tune up α and λ to find the best model by observing the recommendation metrics

Multi-Task RecSys

- Case Study 2: Scalarization in RecSys + Text Tasks

  Li, P., Wang, Z., Ren, Z., Bing, L., & Lam, W. (2017). Neural rating regression with abstractive tips generation for recommendation. In ACM SIGIR 2017.

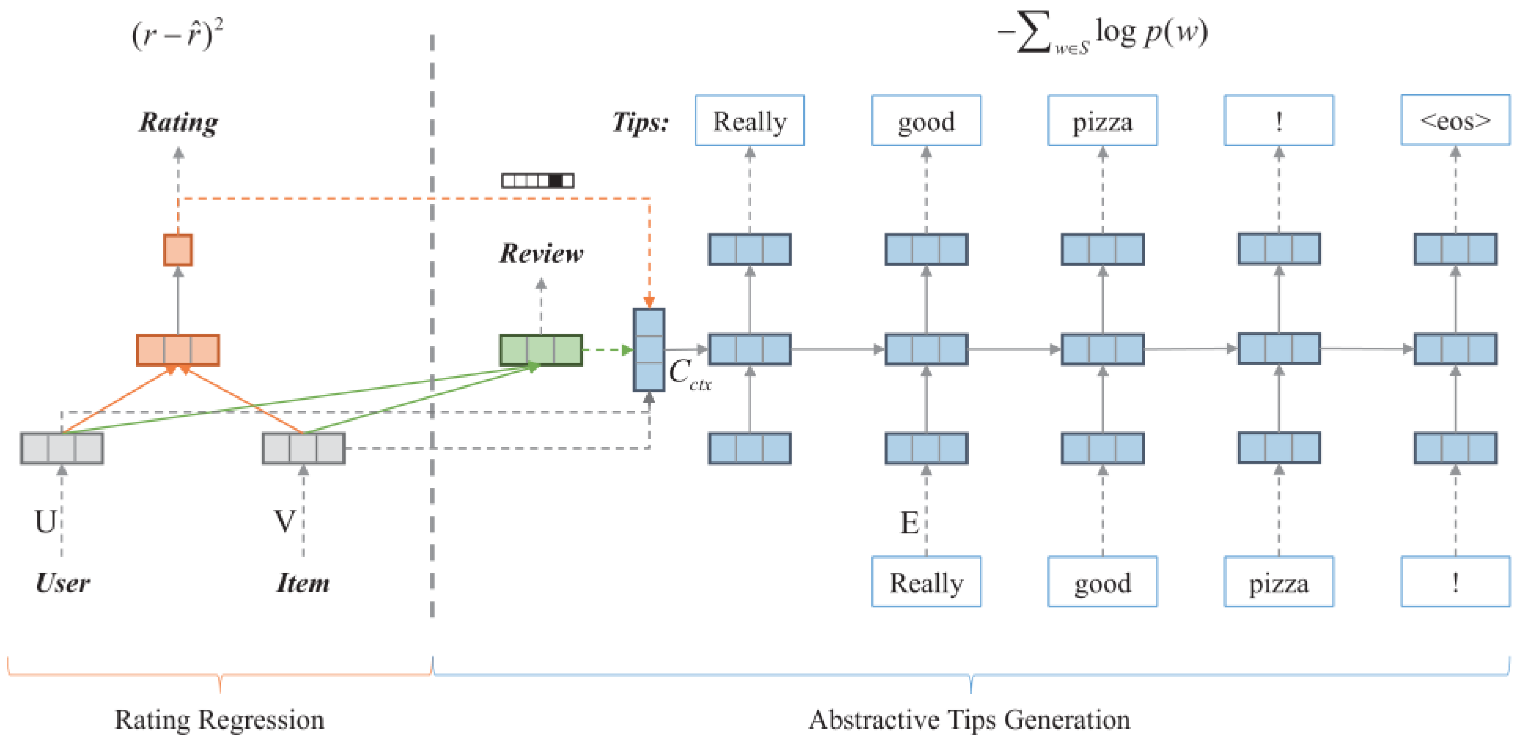
  – Application: optimize rating prediction and generate corresponding tips according to user’s reviews



Multi-Task RecSys

- Case Study 2: Scalarization in RecSys + Text Tasks

Li, P., Wang, Z., Ren, Z., Bing, L., & Lam, W. (2017). Neural rating regression with abstractive tips generation for recommendation. In ACM SIGIR 2017.



Multi-Task RecSys

- Case Study 2: Scalarization in RecSys + Text Tasks

  Li, P., Wang, Z., Ren, Z., Bing, L., & Lam, W. (2017). Neural rating regression with abstractive tips generation for recommendation. In ACM SIGIR 2017.

  – Scalarization in the joint learning process

$$\mathcal{J} = \min_{\mathbf{U},\mathbf{V},\mathbf{E},\Theta} (\lambda_r \mathcal{L}^r + \lambda_c \mathcal{L}^c + \lambda_s \mathcal{L}^s + \lambda_n (\|\mathbf{U}\|_2^2 + \|\mathbf{V}\|_2^2 + \|\Theta\|_2^2))$$

Rating loss
$$\mathcal{L}^r = \frac{1}{2|\mathcal{X}|} \sum_{u \in \mathcal{U}, i \in \mathcal{I}} (\hat{r}_{u,i} - r_{u,i})^2$$

Review loss
$$\mathcal{L}^c = -\sum_{k=1}^{|\mathcal{V}|} \mathbf{c}^{(k)} \log \hat{\mathbf{c}}^{(k)}$$

Tips loss
$$\mathcal{L}^s = -\sum_{w \in Tips} \log \hat{\mathbf{s}}^{(I_w)}$$

Multi-Task RecSys

- Case Study 2: Scalarization in RecSys + Text Tasks

  Li, P., Wang, Z., Ren, Z., Bing, L., & Lam, W. (2017). Neural rating regression with abstractive tips generation for recommendation. In ACM SIGIR 2017.

  – Results: improve both tasks
    - Tune up the weights in the joint loss function
    - Observing metrics for two tasks
      – Rating prediction task $MAE = \frac{1}{N}\sum_{u,i}\left|r_{u,i} - \hat{r}_{u,i}\right|$

      – Tips generation $ROUGE\text{-}N(s) = \sum_{g_n \in s_h} C_m(g_n) / \sum_{g_n \in \tilde{s}} C(g_n)$
      Measuring the overlapping between generated tips and true tips based on the co-occurrence on the n-grams

Multi-Task RecSys

- Summary
  - The nature of a multi-task RecSys is involved a process of joint learning which is a stage of multi-objective optimization
  - Weighted sum is the most common scalarization method in these work

Multi-Task RecSys

- Summary
  - However
    - Tune up parameters as long as it can beat baselines
      But there could be better solutions potentially missed in the solution search

RecSys with MOO

- How to use MOO/its Knowledge in RecSys
  1) Recommendation Task as a MOO Process
     - A MOO is involved in recommendation process
     - MOO techniques introduced by David can be reused to serve in recommender systems
  2) Enhanced RecSys with Dominance Relations
     - There are is multi-objective optimization process
     - Knowledge or skills (e.g., dominance relations) from MOO can be reused to improve recommendations

RecSys with MOO Knowledge

- Case Studies
  - RecSys Using Non-dominated Neighbors
    - Özsoy, et al. "Multi-objective optimization based location and social network aware recommendation." 10th IEEE International Conference on Collaborative Computing. IEEE, 2014.
    - Zheng, Yong. "Non-dominated differential context modeling for context-aware recommendations." Applied Intelligence (2022): 1-14.
  - RecSys Using Pareto Ranking
    - Ribeiro, Marco Tulio, et al. "Multi-objective pareto-efficient approaches for recommender systems." ACM Transactions on Intelligent Systems and Technology (TIST) 5.4 (2014): 1-20.
    - Zheng, Yong, and David Wang. "Multi-Criteria Ranking: Next Generation of Multi-Criteria Recommendation Framework." IEEE Access 10 (2022)

RecSys with MOO Knowledge

- Case Study 1: RecSys Using Non-dominated Neighbors
- Recap: What is dominance relation?

A solution x is said to be Dominated by x^* if and only if

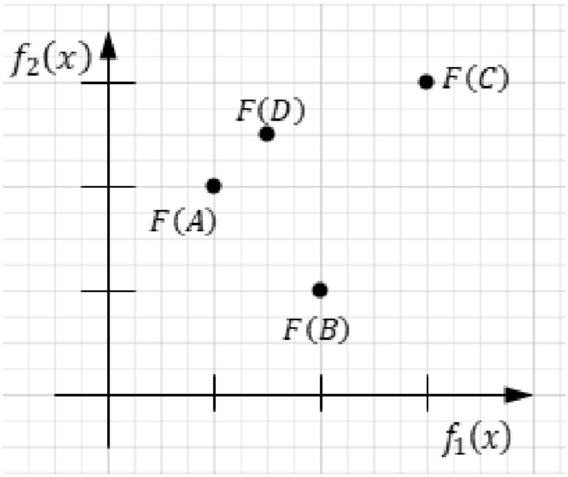
$$f_m(x^*) \leq f_m(x) \text{ for all } m = 1,2,\dots,M$$

and there exists at least one m' such that:

$$f_{m'}(x^*) < f_{m'}(x)$$

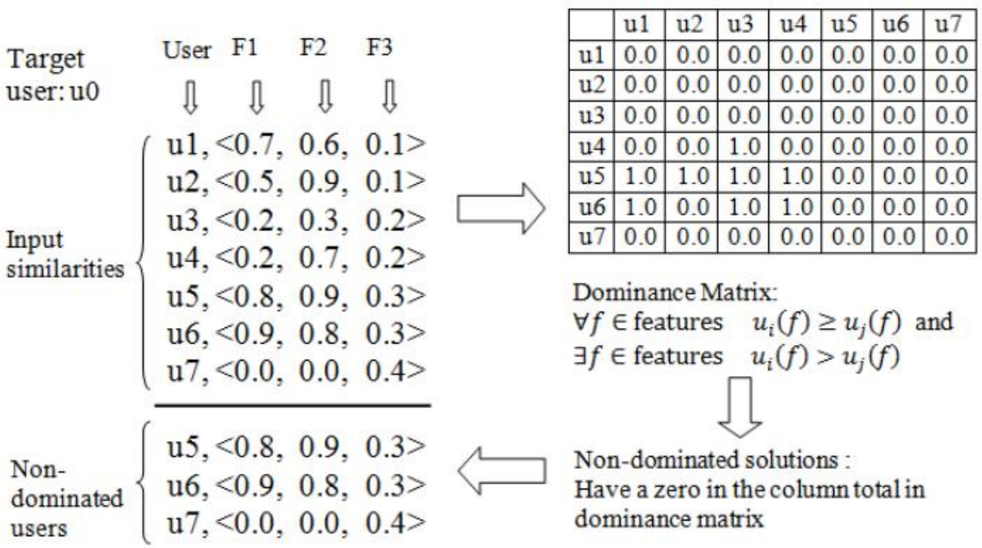
A and B dominate C, D is only dominated by A.
A and B: no dominance relationship
D and B: no dominance relationship

$$\min_x (f_1, f_2)$$



RecSys with MOO Knowledge

- Case Study 1: RecSys Using Non-dominated Neighbors
  - Özsoy, et al. "Multi-objective optimization based location and social network aware recommendation." IEEE Conference on Collaborative Computing. IEEE, 2014.



	u1	u2	u3	u4	u5	u6	u7
u1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
u2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
u3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
u4	0.0	0.0	1.0	0.0	0.0	0.0	0.0
u5	1.0	1.0	1.0	1.0	0.0	0.0	0.0
u6	1.0	0.0	1.0	1.0	0.0	0.0	0.0
u7	0.0	0.0	0.0	0.0	0.0	0.0	0.0

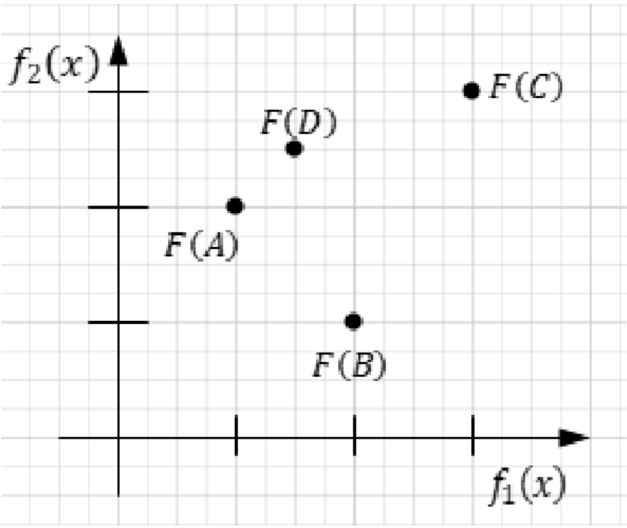
Steps to find Non-Dominated Ngbrs

- Calculate user-user sims based on multiple features (F1, F2, F3,..), e.g., co-ratings, locations, demographic information, etc
- Given a user u0, build a dominance matrix for potential neighbors based on dominance relations
- ND-Ngbrs = users with colsum = 0

$$dom(u,v) = \begin{cases} 1.0 & \forall f\, u(f) \geq v(f) \text{and} \exists f\, u(f) > v(f) \\ 0.0 & \text{otherwise} \end{cases}$$

RecSys with MOO Knowledge

- Case Study 2: RecSys Using Pareto Ranking
- Recap: Pareto ranking in MOEA



Rank (A) = 2, Rank (B) = 1
Rank (D) = 1, Rank (C) = 0

Pareto ranking = sort and rank solutions in MOEA according to the power of dominance

- *Belegundu's ranking*
  Non-dominated solutions with rank 0
  Other solutions with rank 1
- *Goldberg's ranking*
  Rank 1 to non-dominated solutions
  Remove these solutions
  Rank 2 to the non-dominated sols in remaining set
  Remove these solutions
  Rank 3, 4, 5, …..
- *Fonseca & Fleming's ranking (revised/adapted)*
  Rank = number of dominated solutions (example left)

RecSys with MOO Knowledge

- Case Study 2: RecSys Using Pareto Ranking
  - Zheng, Yong, and David Wang. "Multi-Criteria Ranking: Next Generation of Multi-Criteria Recommendation Framework." IEEE Access 10 (2022)

Figure 1: Ratings on OpenTable.com

User	Item	Rating	Food	Service	Ambience
U_1	T_3	4	4	3	4
U_2	T_2	3	3	3	3
U_3	T_1	?	?	?	?

Table 1: Example of Data

Multi-Criteria Recommendation Systems (MCRS)

- We have users' overall and multi-criteria ratings in the data set
- How MCRS works
  - Predict multi-criteria ratings by given a user and an item
  - Aggregate the ratings above to estimate the overall rating

RecSys with MOO Knowledge

- Case Study 2: RecSys Using Pareto Ranking
  - Zheng, Yong, and David Wang. "Multi-Criteria Ranking: Next Generation of Multi-Criteria Recommendation Framework." IEEE Access 10 (2022)

Item	Rating	Food	Service	Ambience
T_1	?	5	5	5
T_2	?	4	4	4
T_3	?	3	3	3
T_4	?	4	3	3
T_5	?	4	5	3.5

Table 2: Item Candidates to be Ranked

Item	Rating	Ranking Score
T_1	?	4
T_2	?	2
T_3	?	0
T_4	?	1
T_5	?	2

Table 3: Ranking Scores for Item Candidates

Ranking score = the number of items that an item *T* can dominate

We can utilize this ranking score to sort and rank items for top-N recommendations

Agenda

01

Intro

Intro to RecSys

02

Why MOO

Contexts in which
we need MOO in RecSys

03

Case Studies

Examples of using
MOO in RecSys

04

Summary

Summary, Guideline,
Challenges & QA

Summary

- Motivations of Using MOO in RecSys
  1) Finding a balance among multiple objectives
     - Considering multiple RecSys metrics
     - Considering objectives from multiple stakeholders
     - Looking for improvements on multiple tasks
     - ….
  2) Improve RecSys by considering multiple objectives
     - Improving Group RecSys by considering diff objectives
     - Improving RecSys by considering objs in pre-processing
     - …

Summary

- How to use MOO/its Knowledge in RecSys
  1) Recommendation Task as a MOO Process
     - A MOO is involved in recommendation process
     - MOO techniques introduced by David can be reused to serve in recommender systems
  2) Enhanced RecSys with Dominance Relations
     - There are is multi-objective optimization process
     - Knowledge or skills (e.g., dominance relations) from MOO can be reused to improve recommendations

1). Recommendation Task as a MOO Process

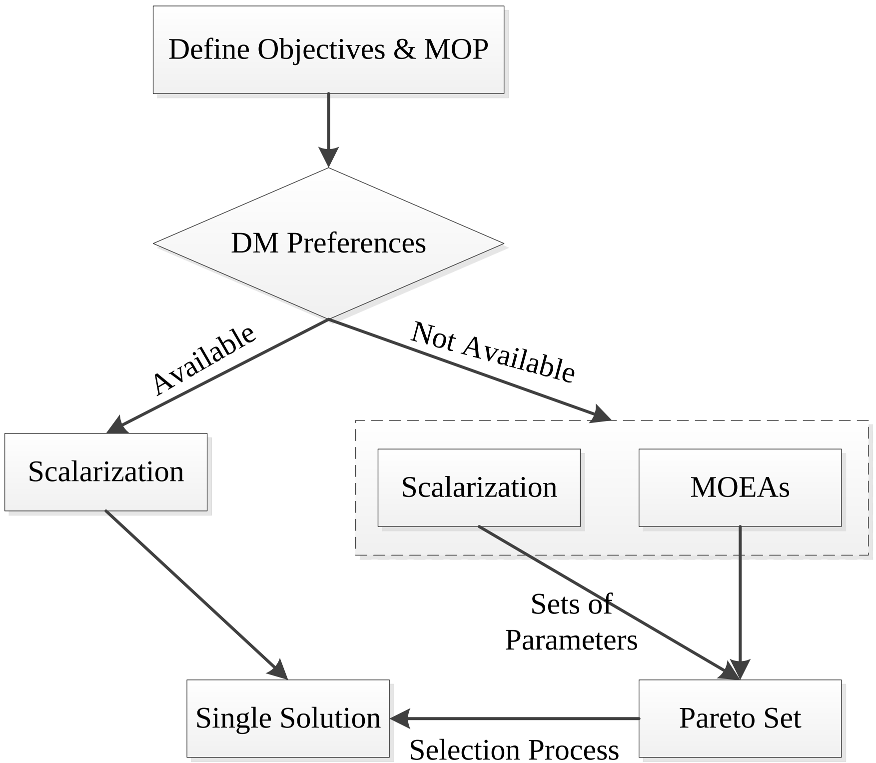
- Multi-objective RecSys utilize MOO to solve the multi-objective problems in RecSys

2). Enhanced RecSys with Dominance Relations

- Examples
  - RecSys Using Non-dominated Neighbors
    - Özsoy, et al. "Multi-objective optimization based location and social network aware recommendation." 10th IEEE International Conference on Collaborative Computing. IEEE, 2014.
    - Zheng, Yong. "Non-dominated differential context modeling for context-aware recommendations." Applied Intelligence (2022): 1-14.
  - RecSys Using Pareto Ranking
    - Ribeiro, Marco Tulio, et al. "Multi-objective pareto-efficient approaches for recommender systems." ACM Transactions on Intelligent Systems and Technology (TIST) 5.4 (2014): 1-20.
    - Zheng, Yong, and David Wang. "Multi-Criteria Ranking: Next Generation of Multi-Criteria Recommendation Framework." IEEE Access 10 (2022)

Summary

- Suggested Workflow



Challenges in RecSys with MOO

- Challenges
  - Scalarization or MOEA?
    Do we have weights on objectives? Usually No
  - Who is the decision maker?
    - The end user
    - Different stakeholders
    - The system designer/developer/ provider of recommendation services?

Challenges in RecSys with MOO

- Using Scalarization (e.g., multi-task)
  - Hyperparameters tuning v.s. Grid/Extensive Search
- More comparisons needed
  - Different ways to select a single optimal solution from the Pareto set
  - Different MOO methods, e.g., weighted sum is the most popular scalarization, how about others?
  - Compare multiple objectives, even if the final goal is to improve RecSys only, e.g., GroupRecSys

Challenges in RecSys with MOO

- How to achieve a balance
  - Offline experiments vs online studies
  - The balance requires user studies
    - Balance among multiple metrics
      What are the acceptable loss in the metrics
    - Balance among multiple stakeholders
      How about the satisfaction by other stakeholders
    - ….

Summary

- Our Tutorial
  - Website: https://moorecsys.github.io/
  - Slide: https://github.com/moorecsys/moorecsys.github.io
  - Supplementary materials:
    - Yong Zheng, David (Xuejun) Wang. "Multi-Objective Recommendations: A Tutorial". CoRR abs/2108.06367. Aug, 2021. https://arxiv.org/abs/2108.06367
    - Yong Zheng, David (Xuejun) Wang. "A Survey of Recommender Systems with Multi-Objective Optimization", Neurocomputing, Elsevier, 2022. https://doi.org/10.1016/j.neucom.2021.11.041

Tutorial: Multi-Objective Recommendations

Yong Zheng, Illinois Institute of Technology, USA
David (Xuejun) Wang, Morningstar, Inc., USA

at the IEEE International Conference on
Data Mining (ICDM) 2022