Comprehensive Fair Meta-learned Recommender System

Tianxin Wei, Jingrui He

twei10@illinois.edu, jingrui@illinois.edu

University of Illinois at Urbana Champaign

ILLINOIS

Roadmap

Background
- Why RecSys is important?
- What are the current challenges?

Problem Definition
- What problem are we solving?
- Situation of the existing work?

Proposed Model
- Preliminary
- Proposed CLOVER Framework

Analysis and Results
- Experimental Effectiveness
- Case study
- Conclusion

Age of Information Explosion

- Serious Issue of **Information Overloading**
  - E.g., there are 800 million videos on Youtube

- Personalized recommender systems have been widely used for **mining user preference** in various web services, such as:
  - E-commerce, e.g., Walmart, Amazon, etc.
  - Social media, e.g., Facebook, LinkedIn, etc.
  - Online entertainment, e.g., Netflix, Tiktok, etc.

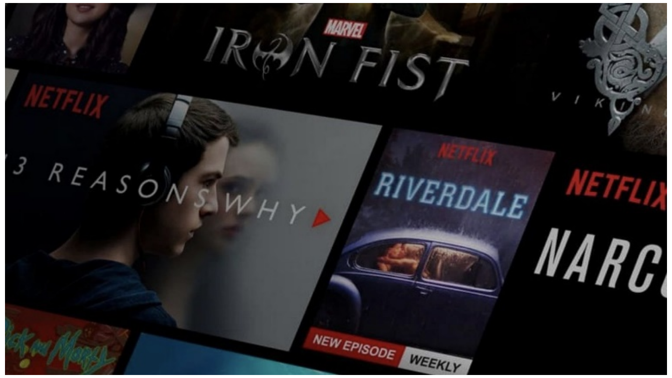
E-commerce Systems

Product Recommendation

Social Networks

Friend Recommendation

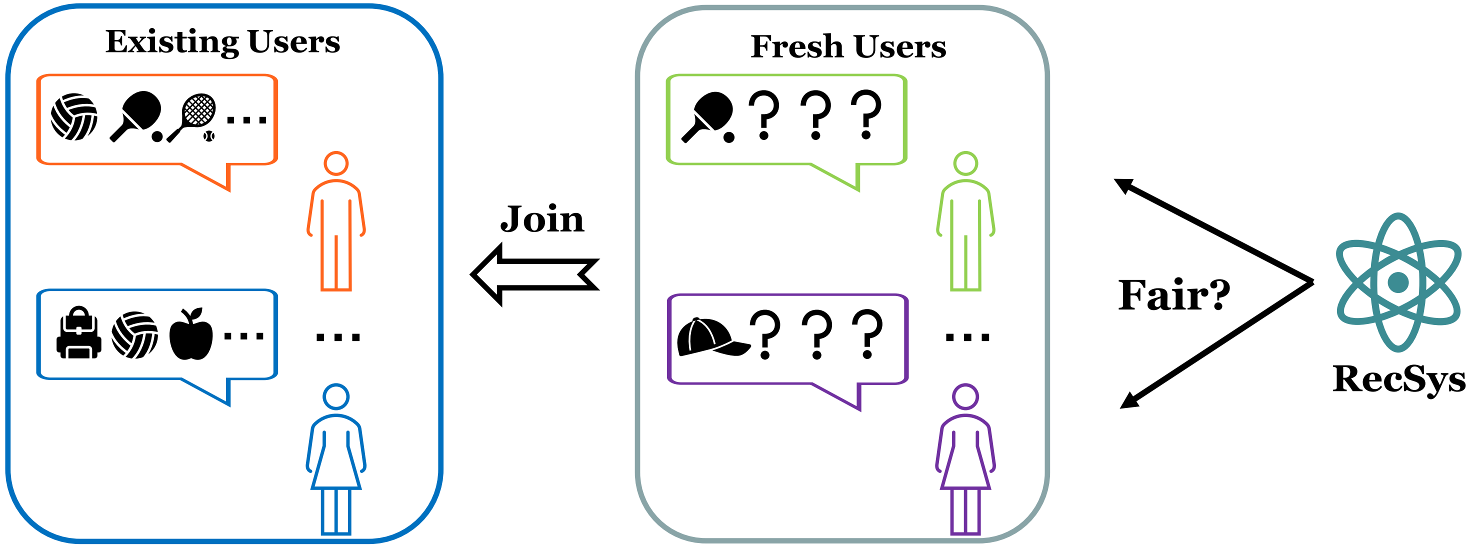
Online Entertainment Systems



Music/Video Recommendation

Cold-start Recommendation

- In recommender systems, one common challenge is the **cold-start problem**, where interactions are very limited for **fresh users** in the systems



- To address this challenge, recently, the **meta-optimization** idea is introduced into the recommendation scenarios
  - They aim at deriving **general knowledge** across various users to **rapidly adapt** to the future new users

- Despite the success of meta-learning at improving the recommendation performance with cold-start, the **fairness issues** are largely **overlooked**

Comprehensive Fairness

- There are several **different definitions of fairness** in recommender system
  - **Individual** Fairness
  - **Group** Fairness
  - **Counterfactual** Fairness

- Previous works mainly consider one perspective of fairness

- How to understand the relationships between these definitions and impose them **comprehensively** remains a challenge

Roadmap

Background

- Why RecSys is important?
- What are the current challenges?

Problem Definition

- What problem are we solving?
- Situation of the existing work?

Proposed Model

- Preliminary
- Proposed CLOVER Framework

Analysis and Results

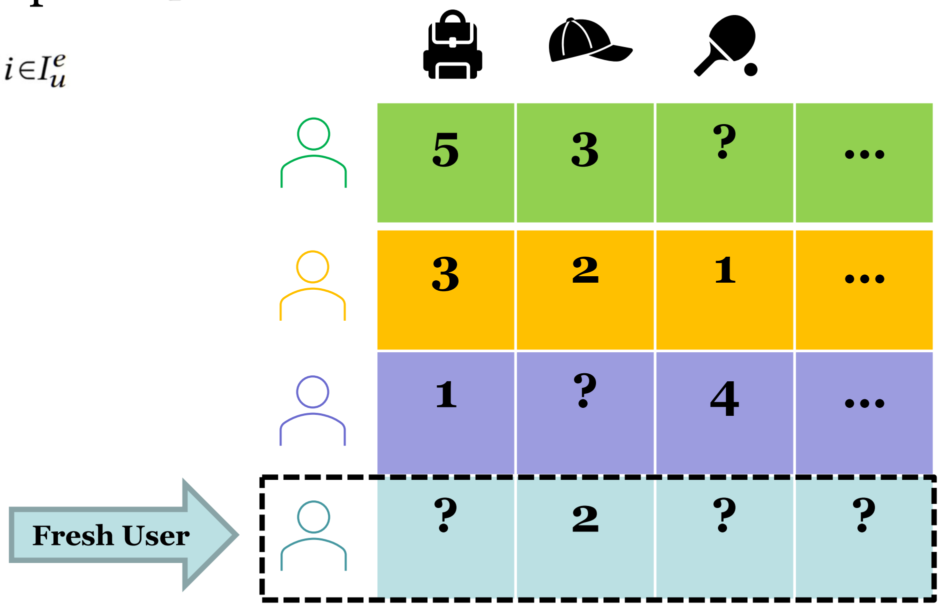
- Experimental Effectiveness
- Case study
- Conclusion

Problem Definition

- **The cold-start problem in recommender system**
  - **Given:**
    - Fresh user u with profile x_u
    - Limited existing interacted items I_u^e along with description p_i
    - Training data can be represented as $D_u^e = \{x_u, p_i, y_{ui}\}_{i \in I_u^e}$
  - **Output:**
    - Personalized model for each user

$$(\theta, D_u^e) \to \theta_u$$

    - Estimated interacting scores for the query items



Existing Work

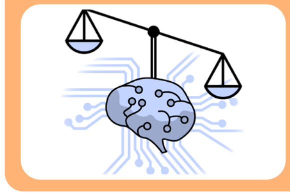
- **Common pitfalls in existing work**
  - Previous works mainly consider **one perspective** of fairness, lack a **comprehensive** understanding of different fairness
  - **Fairness issues** in the **cold-start** recommendation are largely **overlooked**

- **Our designs**
  - Formulate enhancing **comprehensive fairness** as the **multi-task adversarial learning** problem
  - Propose CLOVER to impose fairness in the framework of **meta-learning** with the **interleaving training procedure**

Roadmap

Background
- Why RecSys is important?
- What are the current challenges?

Problem Definition
- What problem are we solving?
- Situation of the existing work?



Proposed Model
- Preliminary
- Proposed CLOVER Framework

Analysis and Results
- Experimental Effectiveness
- Case study
- Conclusion

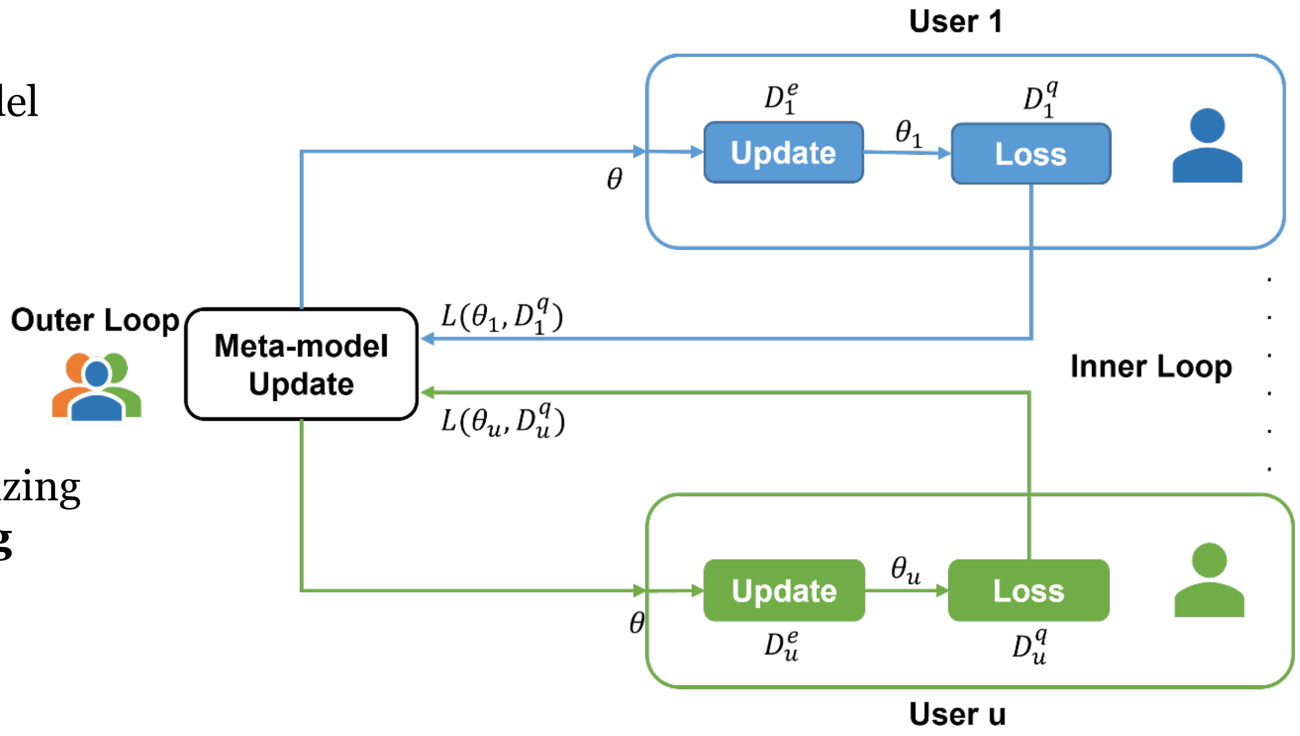
Preliminary: Cold-start Recommender

- **Inner loop**
  - For each user, the framework **initializes** user model with the **latest parameters** of the **meta-model**
  - **Updates** it according to the user's training data

$$(\theta, D_u^e) \longrightarrow \theta_u$$

- **Outer loop**
  - The framework updates the meta-model by minimizing the recommendation loss of user model **regarding each user** to provide a general **initialization**

$$\theta = \theta - \beta \nabla_\theta \sum_{u \in B} L(f_{\theta_u}, D_u^q)$$



Preliminary: Comprehensive Fairness

- The learned representation and recommendation results should **not expose the sensitive information** that correlates with the users **nor show discrimination** towards any **individual** or **group** of users
- The fairness in the recommender system can be reflected from several **different** perspectives as follows:
  - Individual Fairness

Sensitive Attribute

$$IF = \frac{1}{\boxed{|U^f|}} \max_{g} \sum_{u \in U^f} M(\hat{a}_u = \boxed{g}(e_u), \boxed{a_u})$$

Attacker

Test fresh users

  - Group Fairness

Rec Performance

$$GF = \left| \frac{1}{|A_1|} \sum_{\boxed{u_1 \in A_1}} \boxed{R}(u_1) - \frac{1}{|A_2|} \sum_{u_2 \in A_2} R(u_2) \right|$$

User groups

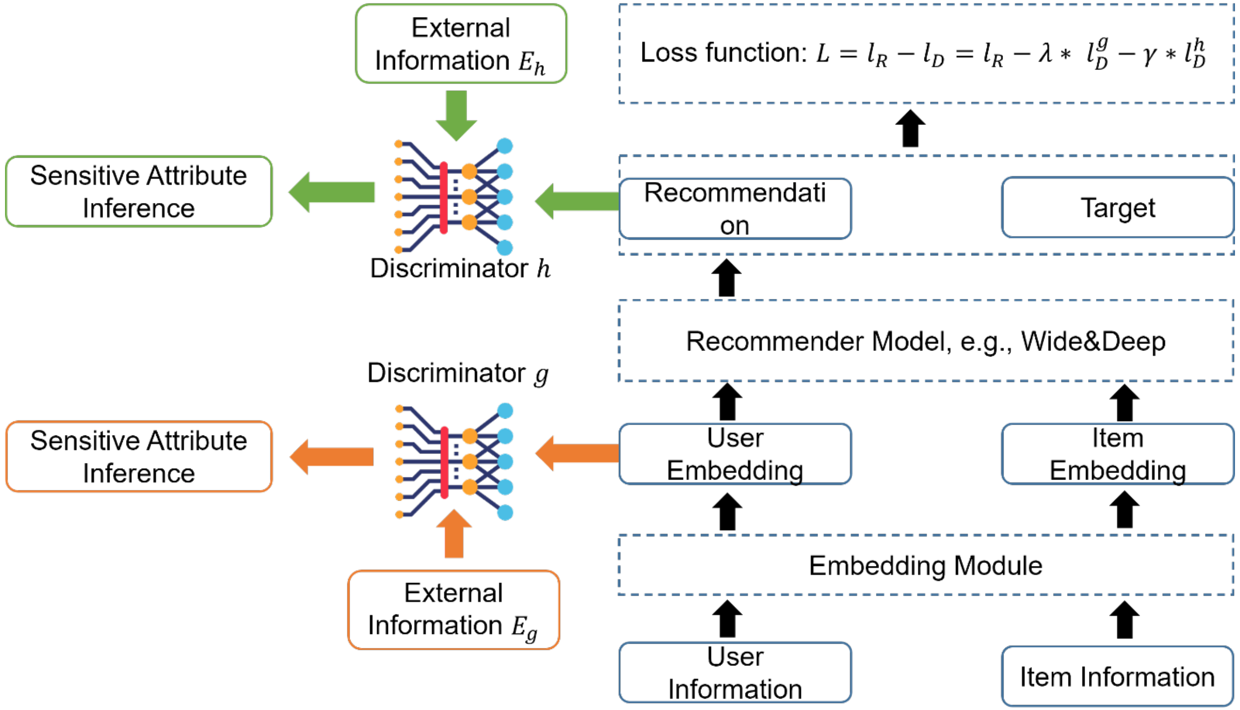
  - Counterfactual Fairness

$$CF = \frac{1}{|U^f|} \sum_{u \in U^f} \left| R(L^u_{a_u} \mid X = x, A = a_u) - R(\boxed{L^u_{a'_u}} \mid X = x, A = a_u) \right|$$

Rec results with counterfactual sensitive attr

Comprehensive Unfairness Mitigation

- In general, we can formulate the unfairness mitigation as the **adversarial learning** problem
- The **discriminator** seeks to optimize its model to **predict the sensitive information**
- The recommender aims to **extract** users' actual **preferences** while **generate fair results** to fulfill specific fairness requirements



$$\min_{\boxed{R}} \max_{D} L = L(f_{\boxed{\theta^r}}, \theta^d) = l_R(f_{\theta^r}) - l_{\boxed{D}}(f_{\theta^r}, \theta^d)$$

Rec　　Parameters of Rec　　Discriminator

Individual Fairness

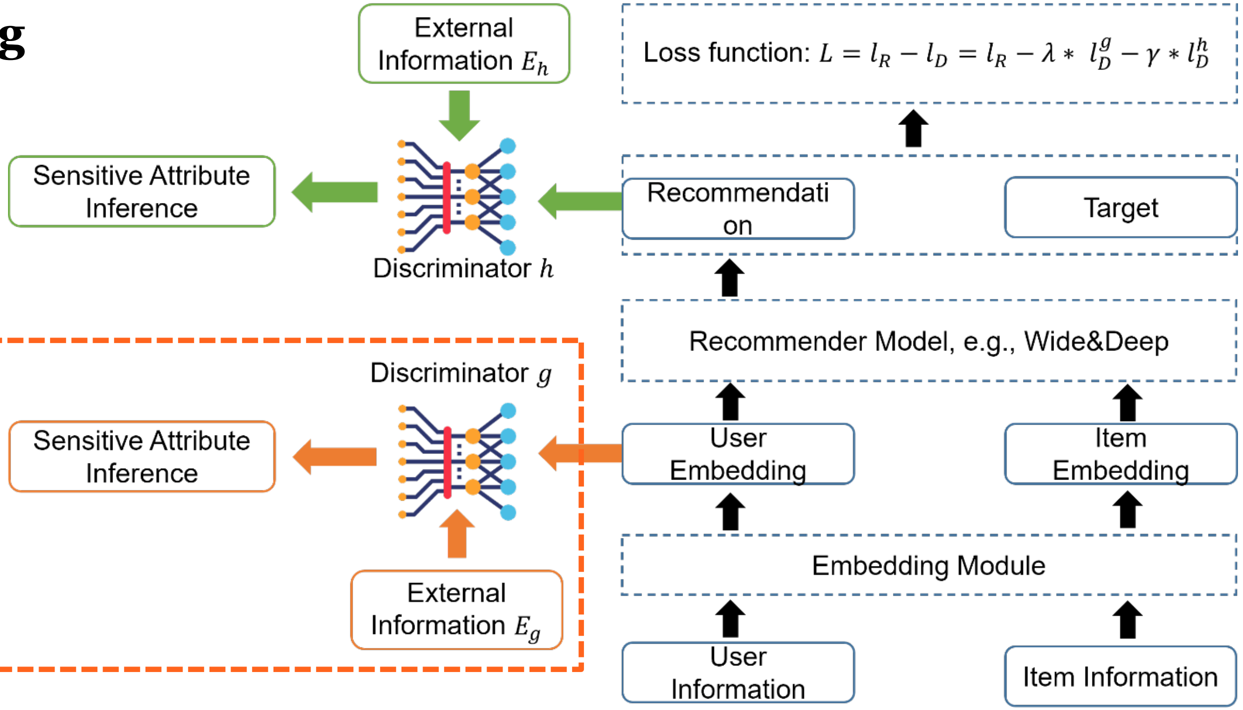
- For individual fairness, it requires the **user modeling process** of the recommender system to **protect** against attacker from inferring the sensitive information

- We conduct to generate the **user embedding** irrelevant to the sensitive information

- The loss can be formulated as:

$$l_D^g = l_D(a_u, \hat{a}_u = g(e_u, \boxed{E_g}))$$

Input external information e.g. target info here can further help optimization



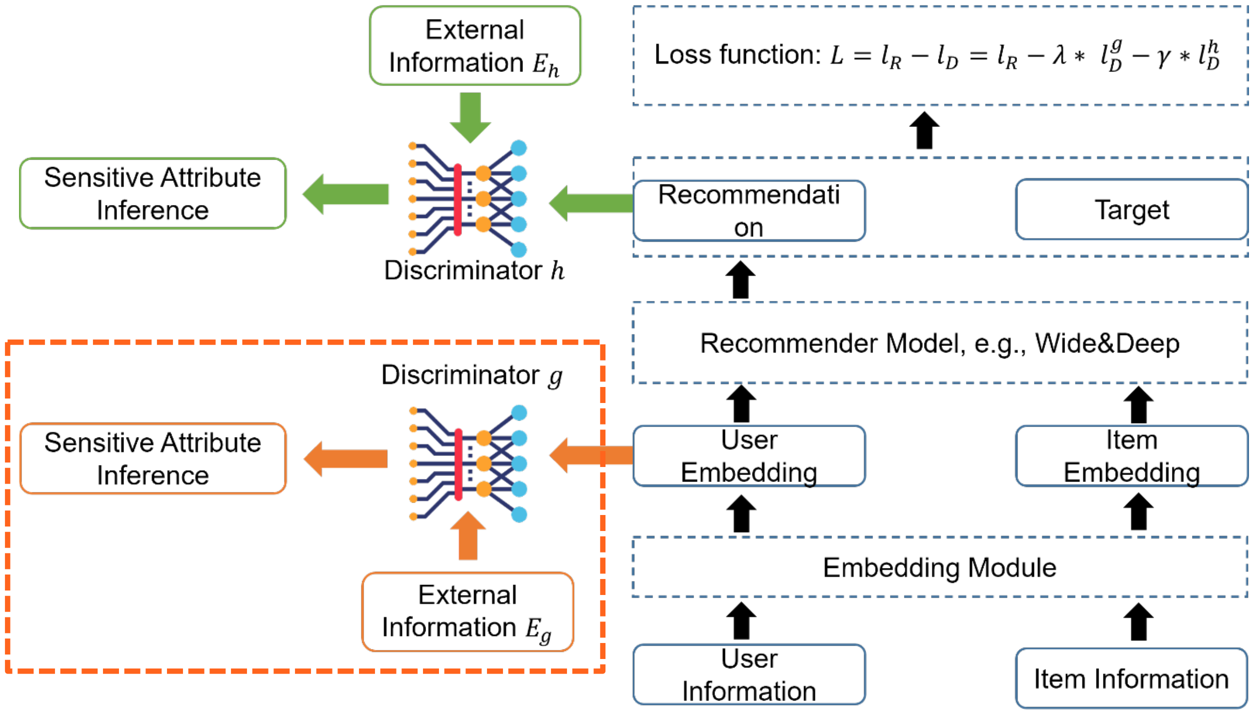
Counterfactual Fairness

- As the user **sensitive attribute** can **only affect** the recommendation results via $a_u \longrightarrow e_u \longrightarrow \hat{y}_{ui}$
- So if we can generate fair user embedding, i.e.,

$$I(a_u; e_u) = 0$$

- Then we can **also satisfy counterfactual fairness**:

$$0 \leq I(a_u; \hat{y}_{ui}) \leq I(a_u; e_u) \ \ \& \ \ I(a_u; e_u) = 0 \Longrightarrow I(a_u; \hat{y}_{ui}) = 0$$



- Thus, **Individual Fairness** ➡ **Counterfactual Fairness**

Group Fairness

- For group fairness, it requires the recommendation **performance** of users to be **identical between different groups**

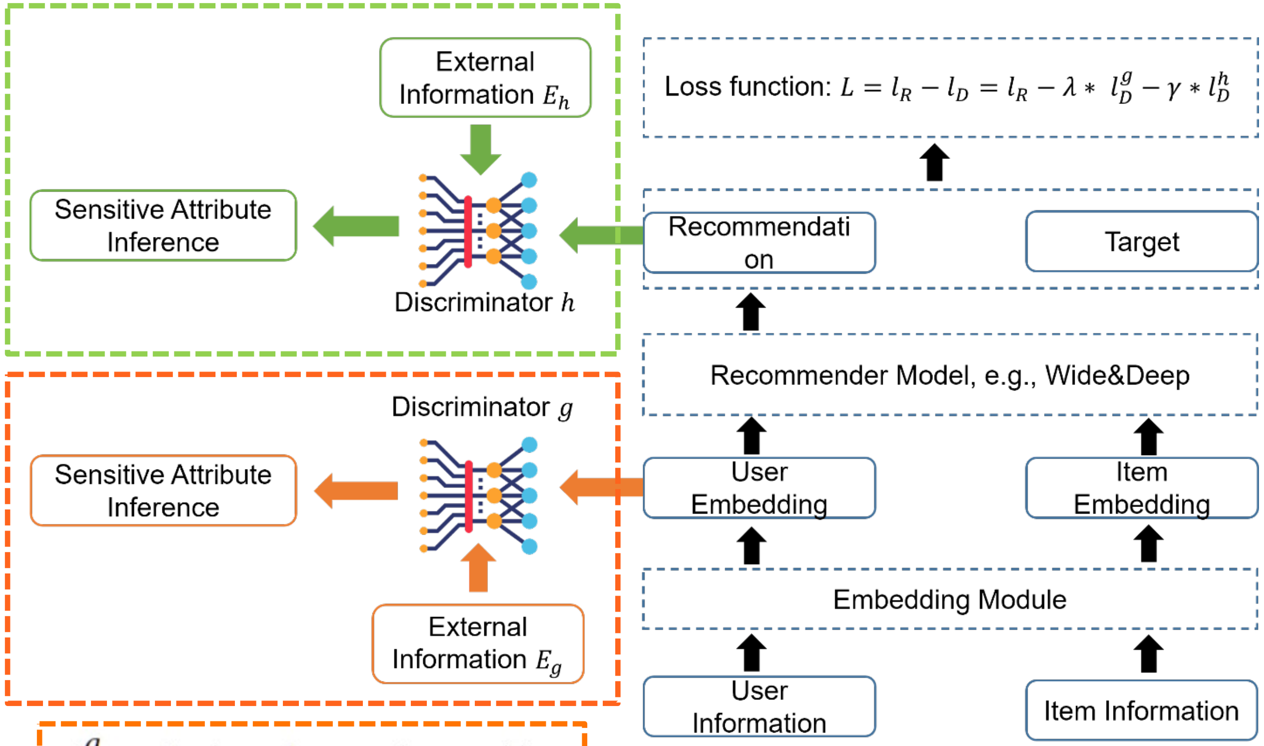
- The goal can then be interpreted as to achieve the **same predicted ratings** across groups **given true rating values**

- The loss for group fairness can be formulated as:

$$\boxed{l_D^h = l_D(a_u, \hat{a}_u = h(y_{ui}, \hat{y}_{ui}, E_h))}$$

- Our final discriminator loss combines them together:

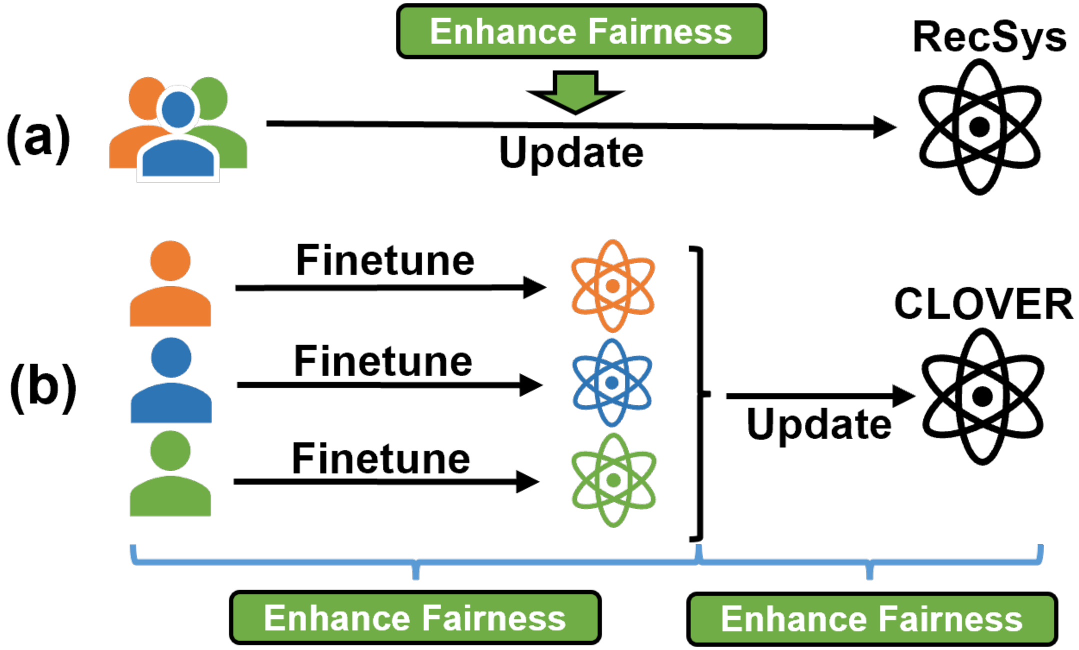
$$l_D = \boxed{\lambda * l_D^g} + \boxed{\gamma * l_D^h}$$



$$\boxed{l_D^g = l_D(a_u, \hat{a}_u = g(e_u, E_g))}$$

Unfairness Mitigation In Cold-start Model

- Our next question is how to **mitigate the unfairness** issues in the **cold-start meta-learned** models?
- Recap that **meta-learning** is a **bi-level** optimization schema consisting of inner and outer loops
- Thus, we'll need to impose fairness (adversarial learning) in **both steps**



Unfairness Mitigation In Cold-start Model

- To better analyze the problem, we first **disentangle** the adversarial recommender training objectives into **three subtasks**:
  - T_r : the task of **recommender loss minimization**, which is required for all recommender models
  - T_1 : the task of **optimizing the discriminator** to predict the sensitive information
  - T_2 : the task of updating the recommender to generate fair results by **fooling the discriminator**
- Our main finding is that: Naively **perform all the subtasks** in **both inner and outer loop** will lead to **suboptimal performance**
- **Inner loop and outer loop require different optimization strategy!**

Unfairness Mitigation In Cold-start Model

- In the **outer loop**, we **perform all the subtasks** to learn a fair recommender initialization

$$\theta^r = \theta^r - \beta \nabla_{\theta^r} \sum_{u \in B} L(f_{\theta_u}) \quad T_r \;\&\; T_2$$

$$\theta^d = \theta^d + \beta \nabla_{\theta^d} \sum_{u \in B} L(f_{\theta_u}) \quad T_1$$

- In the inner loop, we only optimize the recommender loss and the discriminator loss, which has three main explanations:
  - **Model stability**. For fast adaptation, the user model will be fine-tuned by **only a few steps** of gradient descent on limited data, whereas it usually takes **a longer time** for the adversarial game to reach the **desired equilibrium**
  - **Training efficiency**. The **bulk of computation** is largely spent performing **the adversarial game**
  - **Privacy**. Since we no longer need to update the meta-model during deployment, and we **only need to perform task** T_r in testing. In this way, we are **free of user private information after training**

$$\theta_u^r = \theta_u^r - \beta \nabla_{\theta_u^r} l_R(f_{\theta_u}) \quad T_r$$

$$\theta_u^d = \theta_u^d + \beta \nabla_{\theta_u^d} L(f_{\theta_u}) \quad T_1$$

Roadmap

Background

- Why RecSys is important?
- What are the current challenges?

Problem Definition

- What problem are we solving?
- Situation of the existing work?

Proposed Model

- Preliminary
- Proposed CLOVER Framework

Analysis and Results

- Experimental Effectiveness
- Case study
- Conclusion

Experiments

- Experimental Setup
  - Data sets:
    - ML-1M
    - BookCrossing
    - ML-100K
  - Evaluation:
    - Split users with ratio of 70%/10%/20%
    - Test users are not seen during training
  - Baselines:
    - Traditional collaborative filtering
    - Traditional cold-start
    - Meta-learning Algorithm
      - With Fairness Consideration
  - Metrics:
    - MAE
    - NDCG@3
    - AUC (Individual Fairness)
    - CF (Counterfactual Fairness)
    - GF (Group Fairness)

Table 1: Statistics of Datasets.

Dataset	ML-1M	BookCrossing	ML-100K
No. of users	6,040	278,858	943
No. of items	3,706	271,379	1,682
No. of ratings	1,000,209	1,149,780	100,000
Sparsity	95.5316%	99.9985%	93.6953%
User contents	Gender, Age, Occupation, Zip code	Age, Location	Gender, Age, Occupation, Zip code
Item contents	Publication year, Rate, Genre, Director, Actor	Publication year, Author, Publisher	Publication year, Genre
Range of ratings	1 ~ 5	1 ~ 10	1 ~ 5

Experiments

Table 2: Experimental results on the three datasets averaged over five independent runs. Arrows (↑, ↓) indicate the direction of better performance. CLOVER keeps the predictive power of the original recommender model while improving their fairness. Bold values indicate the best performance with regard to the meta-learned recommender model.

Traditional CF

Cold-start

Meta learning

	ML-1M					BookCrossing					ML-100K				
	MAE ↓	NDCG ↑	AUC ↓	CF ↓	GF ↓	MAE ↓	NDCG ↑	AUC ↓	CF ↓	GF ↓	MAE ↓	NDCG ↑	AUC ↓	CF ↓	GF ↓
PPR	0.943±0.012	0.672±0.004	0.932±0.013	0.187±0.058	0.052±0.012	2.374±0.015	0.598±0.005	0.854±0.051	0.194±0.035	0.101±0.013	1.398±0.009	0.486±0.005	0.772±0.036	0.095±0.032	0.051±0.008
Wide&Deep	0.815±0.007	0.691±0.003	0.945±0.024	0.214±0.034	0.055±0.014	1.945±0.028	0.627±0.008	0.833±0.034	0.156±0.029	0.098±0.005	1.256±0.002	0.519±0.006	0.863±0.044	0.112±0.021	0.046±0.012
DropoutNet	0.813±0.005	0.702±0.004	0.965±0.013	0.204±0.021	0.063±0.003	1.855±0.002	0.634±0.005	0.889±0.051	0.194±0.035	0.103±0.011	1.172±0.007	0.544±0.005	0.842±0.036	0.131±0.018	0.044±0.013
NLBA	0.795±0.006	0.701±0.002	0.971±0.024	0.254±0.047	0.056±0.007	1.718±0.007	0.651±0.008	0.943±0.034	0.156±0.029	0.112±0.005	1.213±0.003	0.553±0.006	0.891±0.044	0.129±0.023	0.047±0.004
MELU	0.743±0.008	0.755±0.002	1.000±0.000	0.264±0.058	0.054±0.011	**1.332±0.008**	**0.723±0.006**	1.000±0.000	0.259±0.094	0.103±0.011	0.892±0.009	0.652±0.017	0.974±0.016	0.157±0.032	0.048±0.014
MELU+BS	0.745±0.003	0.744±0.003	1.000±0.000	0.277±0.047	0.049±0.012	1.343±0.005	0.712±0.005	1.000±0.000	0.291±0.058	0.101±0.010	0.891±0.002	0.651±0.011	1.000±0.000	0.174±0.019	0.046±0.008
MELU+Reg	0.762±0.002	0.748±0.002	0.894±0.011	0.145±0.033	0.048±0.009	1.357±0.008	0.713±0.004	0.912±0.023	0.232±0.024	0.097±0.011	0.914±0.007	0.648±0.013	0.873±0.036	0.132±0.012	0.045±0.005
MELU+IPW	0.751±0.012	0.751±0.002	0.975±0.008	0.187±0.013	0.047±0.011	1.365±0.004	0.707±0.006	1.000±0.000	0.245±0.044	0.096±0.007	0.903±0.008	0.643±0.005	0.946±0.024	0.117±0.038	0.047±0.009
MELU+MACR	0.744±0.005	0.750±0.004	0.878±0.031	0.116±0.027	0.051±0.011	1.352±0.006	0.715±0.002	0.867±0.016	0.179±0.094	0.097±0.013	0.887±0.011	0.656±0.002	0.815±0.019	0.108±0.016	0.044±0.007
CLOVER	**0.731±0.005**	**0.756±0.005**	**0.632±0.057**	**0.044±0.014**	**0.040±0.006**	1.352±0.006	0.722±0.006	**0.546±0.032**	**0.027±0.011**	**0.089±0.009**	**0.880±0.009**	**0.666±0.003**	**0.562±0.021**	**0.032±0.015**	**0.036±0.006**

- Compared with baselines, **meta-learning strategy** has a **better recommendation performance** while **worse fairness performance**

- Our proposed CLOVER substantially **outperforms all baseline** methods concerning the fairness performance while not sacrificing the recommendation performance

Case Study

- **Ablation study, generalization ability and hyperparameter Study**

Table 3: Effect of different optimization objectives.

	MAE ↓	NDCG ↑	AUC ↓	CF ↓	GF ↓
MELU	$0.743_{\pm 0.008}$	$0.755_{\pm 0.002}$	$1.000_{\pm 0.000}$	$0.264_{\pm 0.058}$	$0.054_{\pm 0.011}$
CLOVER w/o	$0.739_{\pm 0.004}$	$0.754_{\pm 0.003}$	$1.000_{\pm 0.000}$	$0.332_{\pm 0.069}$	$0.056_{\pm 0.013}$
CLOVER$_{T2}$	$0.761_{\pm 0.007}$	$0.741_{\pm 0.004}$	$1.000_{\pm 0.000}$	$0.247_{\pm 0.042}$	$0.062_{\pm 0.007}$
CLOVER$_{T1\&T2}$	$0.749_{\pm 0.005}$	$0.751_{\pm 0.008}$	$0.792_{\pm 0.035}$	$0.105_{\pm 0.015}$	$0.051_{\pm 0.008}$
CLOVER	$\mathbf{0.731}_{\pm 0.005}$	$\mathbf{0.756}_{\pm 0.005}$	$\mathbf{0.632}_{\pm 0.057}$	$\mathbf{0.044}_{\pm 0.014}$	$\mathbf{0.040}_{\pm 0.006}$

Table 4: Effect of different adversarial loss functions.

	MAE ↓	NDCG ↑	AUC ↓	CF ↓	GF ↓
MELU	$0.743_{\pm 0.008}$	$0.755_{\pm 0.002}$	$1.000_{\pm 0.000}$	$0.264_{\pm 0.058}$	$0.054_{\pm 0.011}$
CLOVER w/o E^h	$0.735_{\pm 0.005}$	$\mathbf{0.758}_{\pm 0.002}$	$0.667_{\pm 0.041}$	$0.051_{\pm 0.009}$	$0.044_{\pm 0.006}$
CLOVER w/o l_D^h	$\mathbf{0.730}_{\pm 0.004}$	$0.754_{\pm 0.006}$	$0.687_{\pm 0.062}$	$0.067_{\pm 0.013}$	$0.055_{\pm 0.010}$
CLOVER w/o E^g	$0.736_{\pm 0.006}$	$0.752_{\pm 0.003}$	$0.725_{\pm 0.037}$	$0.056_{\pm 0.015}$	$0.049_{\pm 0.012}$
CLOVER w/o l_D^g	$0.738_{\pm 0.005}$	$0.754_{\pm 0.008}$	$0.914_{\pm 0.018}$	$0.198_{\pm 0.021}$	$0.043_{\pm 0.008}$
CLOVER	$0.731_{\pm 0.005}$	$0.756_{\pm 0.005}$	$\mathbf{0.632}_{\pm 0.057}$	$\mathbf{0.044}_{\pm 0.014}$	$\mathbf{0.040}_{\pm 0.006}$

Table 5: Generalization ability of CLOVER.

	MAE ↓	NDCG ↑	AUC ↓	CF ↓	GF ↓
MELU	$0.743_{\pm 0.008}$	$0.755_{\pm 0.002}$	$1.000_{\pm 0.000}$	$0.264_{\pm 0.058}$	$0.054_{\pm 0.011}$
CLOVER$_{MELU}$	$\mathbf{0.731}_{\pm 0.005}$	$\mathbf{0.756}_{\pm 0.005}$	$\mathbf{0.632}_{\pm 0.057}$	$\mathbf{0.044}_{\pm 0.014}$	$\mathbf{0.040}_{\pm 0.006}$
MetaCS	$0.721_{\pm 0.005}$	$\mathbf{0.776}_{\pm 0.003}$	$1.000_{\pm 0.000}$	$0.245_{\pm 0.058}$	$0.061_{\pm 0.015}$
CLOVER$_{MetaCS}$	$\mathbf{0.720}_{\pm 0.002}$	$0.775_{\pm 0.005}$	$\mathbf{0.578}_{\pm 0.037}$	$\mathbf{0.029}_{\pm 0.009}$	$\mathbf{0.044}_{\pm 0.008}$
MAMO	$0.717_{\pm 0.003}$	$0.781_{\pm 0.004}$	$1.000_{\pm 0.000}$	$0.331_{\pm 0.058}$	$0.057_{\pm 0.013}$
CLOVER$_{MAMO}$	$\mathbf{0.711}_{\pm 0.004}$	$\mathbf{0.786}_{\pm 0.002}$	$\mathbf{0.612}_{\pm 0.044}$	$\mathbf{0.036}_{\pm 0.017}$	$\mathbf{0.039}_{\pm 0.011}$

Table 6: Effect of hyper-parameter λ.

	MAE ↓	NDCG ↑	AUC ↓	CF ↓	GF ↓
1e-2	$0.735_{\pm 0.002}$	$0.755_{\pm 0.003}$	$0.872_{\pm 0.056}$	$0.092_{\pm 0.019}$	$0.050_{\pm 0.011}$
1e-1	$0.733_{\pm 0.004}$	$0.754_{\pm 0.004}$	$0.679_{\pm 0.052}$	$0.067_{\pm 0.009}$	$0.045_{\pm 0.007}$
1	$\mathbf{0.731}_{\pm 0.005}$	$\mathbf{0.756}_{\pm 0.005}$	$\mathbf{0.632}_{\pm 0.057}$	$\mathbf{0.044}_{\pm 0.014}$	$\mathbf{0.040}_{\pm 0.006}$
5	$0.737_{\pm 0.007}$	$0.754_{\pm 0.004}$	$0.722_{\pm 0.043}$	$0.079_{\pm 0.021}$	$0.048_{\pm 0.006}$

Table 7: Effect of hyper-parameter γ.

	MAE ↓	NDCG ↑	AUC ↓	CF ↓	GF ↓
1e-2	$0.733_{\pm 0.003}$	$0.754_{\pm 0.004}$	$0.647_{\pm 0.046}$	$0.055_{\pm 0.013}$	$0.044_{\pm 0.011}$
1e-1	$\mathbf{0.731}_{\pm 0.005}$	$\mathbf{0.756}_{\pm 0.005}$	$\mathbf{0.632}_{\pm 0.057}$	$\mathbf{0.044}_{\pm 0.014}$	$\mathbf{0.040}_{\pm 0.006}$
1	$0.734_{\pm 0.005}$	$0.754_{\pm 0.003}$	$0.651_{\pm 0.055}$	$0.060_{\pm 0.020}$	$0.047_{\pm 0.006}$
5	$0.741_{\pm 0.003}$	$0.754_{\pm 0.004}$	$0.941_{\pm 0.017}$	$0.176_{\pm 0.023}$	$0.061_{\pm 0.008}$

Conclusion

- In this paper, we present the **first fairness view** for the **meta-learned recommender systems** with cold-start

- We propose the concept of **comprehensive fairness**, and formulate it as an adversarial learning problem

- A carefully designed framework, CLOVER, is proposed to enable fair representation learning in the meta-learned recommender system

- Extensive experiments on three real-world data sets demonstrate the **effectiveness** of our method, which outperforms a set of strong baselines

THANK YOU!