Adversarial Preference Learning with Pairwise Comparisons

Zitai Wang[1,2], Qianqian Xu[3], Ke Ma[1,2], Yangbangyan Jiang[1,2],
Xiaochun Cao[1,2], Qingming Huang[3,4,5*]
[1]State Key Laboratory of Information Security, Institute of Information Engineering, CAS, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3]Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, CAS, Beijing, China
[4]School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing, China
[5]Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, Beijing, China
wang22ti@gmail.com,xuqianqian@ict.ac.cn,{make,jiangyangbangyan,caoxiaochun}@iie.ac.cn,qmhuang@ucas.ac.cn

ABSTRACT

When facing rich multimedia content and making a decision, users tend to be overwhelmed with redundant options. Recommendation system can improve the users' experience by predicting the possible preference of a given user. The vast majority of the literature adopts the collaborative framework, which relies on a static and fixed formulation of the rating score prediction function (in most cases an inner product function). However, such a static learning paradigm is not consistent with the dynamic feature of human intelligence. Motivated by this, we present a novel adversarial framework for collaborative ranking. On one hand, we leverage a deep generator to approximate an arbitrary continuous score function in terms of pairwise comparison. On the other hand, a discriminator provides personalized supervision signals with increasing difficulty. Different from the traditional static learning framework, our proposed approach enjoys a dynamic nature and unifies both the generative and the discriminative model for collaborative ranking. Comprehensive empirical studies on three real-world datasets show significant improvements of the adversarial framework over the state-of-the-art methods.

KEYWORDS

Generative Adversarial Networks, Collaborative Ranking, Pairwise Comparison

ACM Reference Format:

Zitai Wang, Qianqian Xu, Ke Ma, Yangbangyan Jiang, Xiaochun Cao, Qingming Huang. 2019. Adversarial Preference Learning with Pairwise Comparisons. In *MM '19: 2019 ACM Multimedia Conference, Oct. 21–25, 2019, Nice, France.* ACM, New York, NY, USA. 9 pages. https://doi.org/10.1145/3343031.3350919

*Corresponding author.

MM '19, October 21–25, 2019, Nice, France

ACM ISBN 978-1-4503-6889-6/19/10…$15.00
https://doi.org/10.1145/3343031.3350919

1 INTRODUCTION

Confronting the era of information, online multimedia content (*e.g.,* short videos, movies, music) has been widely accessible to users. Complex options always lead people to confusion when making a decision (we are constantly annoyed with choosing which movie to see, picking out our favorite album out of hundreds of candidates). As a way to alleviate such a dilemma, Recommendation System (RS) gains rising attention in recent years by leveraging a choice prediction scheme [14, 23]. Hitherto, the wisdom of RS has successfully permeated all aspects of our life on the Internet via providing accurate online shopping nevigations [4, 51], making interesting movie/music recommendations [3, 21], and facilitating the online social community [44, 50].

The success of RS has greatly encouraged the relevant researches on user preference prediction models. Among such researches, the Collaborative Filtering (CF) framework is known to be one of the most popular and effective recommendation representatives. The key idea behind CF is that users sharing similar historical behaviors tend to hold similar preference opinions. However, CF methods only consider the preference prediction problem in a point-wise manner. In other words, CF methods only focus on separating the relevant/positive items from the irrelevant/negative items. Practically, the performance of an RS is not measured by point-wise rating prediction, but rather by the relative ranking quality of the top positive items. This directly gives birth to the pair-wise ranking-based variants of CF, which are often referred to as Collaborative Ranking (CR) [2] in the community. Collaborative ranking (CR) is proposed to minimize a ranking loss and predict the order of items. Early collaborative ranking methods realize this goal by simply transforming the pairwise ranking framework to their point-wise counterparts [19, 37, 41, 45]. However, the notorious ***calibration*** drawback further restricts the success of this solution [2]. In one word, the ***calibration*** phenomena refer to the fact that different users might have different sensitivities toward the scale of scores (e.g. Some might find a rating of 4 comparable with that of 5, but others might regard this as a significant difference). Therefore, recent CR variants proceed a step further to exploit pairwise comparisons and optimize the ranking of the scores directly [31, 35, 46, 49].

Existing pairwise CR methods fall into two categories: the generative model and the discriminative model. Given a score function, the generative method directly learns the joint probability distribution of score difference of an item pair and the comparison label [30, 33, 45]. Meanwhile, the discriminative model learns the conditional probability distribution of the comparison label given the score difference. However, both the generative and the discriminative frameworks rely on a static learning paradigm with a fixed formulation of the score function [31, 46]. However, human intelligence is often benefited from its highly dynamic nature. Specifically, the dynamic feature could be interpreted as learning against increasing difficulty and learning against adversarial attacks and noises. Borrowing the strength of human intelligence, our goal in this paper is then to embrace such dynamics in our proposed method. More specifically, our main contributions are listed as follows:

- We propose a novel adversarial learning framework for collaborative ranking to learn with a dynamic scenario. On one hand, a discriminator is deployed to provide increasing difficulty for the model. On the other hand, a generator simultaneously learns a proper score function and the representations of users and items against the discriminator.
- In the core of our framework lies a mini-max game. When the game reaches equilibrium, the resulting generator could automatically capture a proper score function and is capable of resisting hard and noisy examples.
- Theoretical analysis shows that the proposed framework is generic and can express the generative model for collaborative ranking.

2 RELATED WORK

2.1 Collaborative Ranking

Collaborative ranking (CR) is the pair-wise ranking-based variants of collaborative filtering (CF) [34, 38]. Typical CF methods are measured by error metrics like mean squared error (MSE) [15, 18, 32, 36]. However, equally calculating the bias for each feedback could limit its performance, especially when only a few (top) items are recommended to each user [9]. Instead, collaborative ranking is proposed to minimize a ranking loss based on accuracy metrics (*e.g.,* precision and normalized discounted cumulative gain (NDCG)). Essentially, collaborative ranking methods attempt to recommend unobserved items under the ranking framework [2].

To realize this goal, early collaborative ranking methods simply transform the pairwise ranking framework to their point-wise counterparts. CofiRank [45] applies maximum margin matrix factorization [37] to collaborative ranking, and acts as a common baseline method for this task. Balakrishnan and Chopra [2] and Volkovs and Zemel [41] consider collaborative ranking as a learning-to-rank problem [24] and apply existing algorithms to solve it. Decoupled Collaborative Ranking [16] treats the rating as ordinal and emphasizes positively rated items to improve the ranking performance. Lee et al. [19] assumes the rating matrix is locally low-rank and proposes a combination of low-rank matrix models by weighted sum.

To avoid calibration drawback, recent studies shift more attention to pairwise comparisons and model the observations from various perspectives. Rendle et al. [33] are the first to adopt this setting in collaborative ranking and present an optimization criterion based on Bayesian theory and low-rank structure. Shi et al. [35] learn from latent feedback. Yi et al. [49] adopt the matrix completion theory and optimize a convex objective function with the hinge loss. Then, Park et al. [31] propose an effective large-scale non-convex implementation called AltSVM. Furthermore, Wu et al. [46] reduce the time complexity by cleverly simplifying the calculation of gradient and Hessian vector product, leading to its validity on the dataset with massive scale. From another perspective, Xu et al. [47] consider high-order distances (*i.e.*, the bias of the rating difference of each user across different items) in the objective function.

Different from previous work relying on a static learning paradigm, we model the observations from the perspective of adversarial learning and representation learning theory, to enjoy a dynamic nature and better preserve the ranking of items.

2.2 Generative Adversarial Nets

Inspired by the dynamic feature of human intelligence, Generative Adversarial Nets [10] was proposed to generate realistic images. In this framework, an agent generates fake images from noise signals, while its opponent agent discriminates whether the image is synthetic. To improve the performance in conditional generation tasks, Conditional GAN (CGAN) [26] modifies the generator so as to generate samples conditioned on class labels. Later, GAN is incorporated with various learning paradigms [1, 25, 29] and has been widely applied in various fields, such as computer vision [7, 12, 20, 52], natural language processing [22, 39] and so on.

Recently, its application on recommendation systems is gaining increasing attention, but most work primarily focuses on modeling auxiliary information [5, 43]. IRGAN [42] first proposes an adversarial framework for collaborative filtering, where the generator learns to model the preference distribution under the increasingly strict supervision from the discriminator. Such a novel method still suffers from a few deficiencies: (1) Simply classifying numerical feedback into positive and negative categories oversimplifies the ranking of items and thus limits its performance in the top-K recommendation task. (2) The non-differentiable framework has to be optimized by policy gradient descent based on reinforcement learning [27, 40], which is inefficient and time-consuming. (3) Linear score function (*i.e.*, inner product) is too simple to provide sufficient expression ability [13]. Besides, CFGAN [6] adopts a generator to calculate a vector to represent the purchase behavior of each user, and its discriminator decides the purchase vectors are real or not. Compared with our framework, CFGAN could not handle the pairwise data and utilize the advantage of ranking. Moreover, the scalability of CFGAN would be inferior as the length of the purchase vector must equal to the number of items.

3 METHODOLOGY

3.1 Preliminary

Here we introduce the notations and then formulate the traditional collaborative ranking methods mathematically.

Notation. In this paper, n_{user} and n_{item} denote the number of users and items, respectively. $[n]$ represents the set $\{1, \ldots, n\}$. We represent users and items in the same metric space and the corresponding distance can measure the preference of each user. Without loss of generality, let $\boldsymbol{v}_u \in \mathbb{R}^{d_{\text{user}}}$, $\boldsymbol{v}_i \in \mathbb{R}^{d_{\text{item}}}$ be the representations of user u and item i. Furthermore, $s : \mathbb{R}^{d_{\text{user}}} \times \mathbb{R}^{d_{\text{item}}} \mapsto \mathbb{R}$ is the existing but unknown score function of user u and item i which measures the distance between $\boldsymbol{v}_u$ and $\boldsymbol{v}_i$. We note $s_{ui} = s(\boldsymbol{v}_u, \boldsymbol{v}_i)$ and $s_{ui} > s_{uj}$ means user u prefers item i to item j. The training

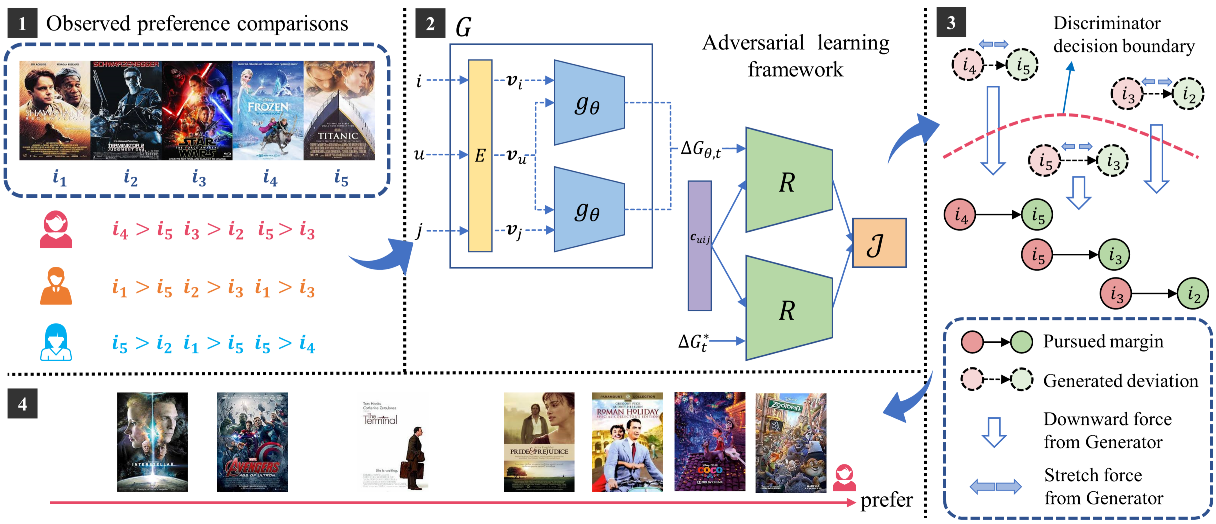

Figure 1: The framework of our method: (1) Our dataset consists of pairwise preference comparisons between different items. (2) To learn a proper score function in a dynamic and adpative manner, we present a novel adversarial framework called CRGAN. (3) Specifically, we design a new discriminative model to effectively enlarge the score difference between a comparison item pair. (4) With the learned scores, we could predict a score for each unseen item preseving the ranking order for the convinience of recommendation.

set $\mathcal{T} = \{(u,i,j)|u \in [n_{\text{user}}], i,j \in [n_{\text{item}}], i \neq j\} \subset [n]^3$ contains all the observed comparisons [31, 48]. For each comparison (u,i,j),

$$y_{uij} = \begin{cases} 1, & \text{if } s_{ui} > s_{uj}, \\ -1, & \text{otherwise,} \end{cases}$$

denotes the label and $\mathcal{Y} = \{y_{uij}|(u,i,j) \in \mathcal{T}\}$ is the label set.

Typically, two types of traditional methods have been proposed to estimate the rankings of multiple users on multiple items.

Generative model. Given the score function s, this model directly learns joint probability distribution $\mathcal{P}(\mathcal{T}, \mathcal{Y})$. In other words, it tries to estimate the preference distribution by minimizing a loss function as follows:

$$\min_{\Theta_g} \sum_{(u,i,j) \in \mathcal{T}} \mathcal{L}_g \left(\mathcal{P} \left(\sigma(s_{ui} - s_{uj}), y_{uij} | \Theta_g \right) \right) \tag{1}$$

where $\sigma : \mathbb{R} \mapsto [0,1]$ represents a generalized linear function; $\mathcal{L}_g : \mathbb{R} \mapsto \mathbb{R}^+$ is a monotonically non-decreasing loss function; Θ_g is the parameters.

Discriminative model. These methods learn the conditional probability distribution $\mathcal{P}(\mathcal{Y}|\mathcal{T})$ with a predefined score function s. The corresponding optimization problem can be formulated as

$$\min_{\Theta_d} \sum_{(u,i,j) \in \mathcal{T}} \mathcal{L}_d \left(\mathcal{P} \left(y_{uij} | \sigma(s_{ui} - s_{uj}), \Theta_d \right) \right). \tag{2}$$

3.2 Adversarial Learning Framework

As mentioned above, the score function s is essential in the collaborative ranking task. With a well-defined score function, both two models can be adopted to estimate the preference of multiple users. However, defining a good match score function is so challenging that the usual metrics, *e.g.,* ℓ_2 norm, cosine similarity, Kullback–Leibler divergence or simple linear model, have been proved that they are not suitable for the large-scale collaborative filtering system. Without prior knowledge of the score function s, we leverage a deep generative model to approximate the probabilistic computations of preferences distribution. This generative model simultaneously obtains the unified user-item representation and a non-linear score function. It is noteworthy that the ranking of the scores is much essential than the score values themselves in the recommendation task. Here we incorporate two generators which estimate the score difference between different items. Moreover, a deep discriminative model maps the feature of each comparison (*i.e.*, the code containing the information of the user and item) and the difference of learned score pairs to a class label. This discriminative model forces the score difference of each pair to approach the pursued margin (*i.e.*, the margin of the optimal classifier for deciding the label of a triplet (u,i,j)) as much as possible. In a word, we propose an adversarial ranking framework, where the generative model is pitted against an adversary: a discriminative model that learns to determine whether the score difference is the pursued margin or not. Such competition between the two agents could generate the ranking of items based on user preference and improves the overall performance.

Generator. We first present the generative model, which learns the score function and the representations of user-item at the same time. To handle the complicated interaction and dissect the relevance between users and items, the non-linear score function could be considered. The most popular choice is neural networks, which are able to approximate any continuous function. Then, we denote the generator as:

$$g_\theta(u,i) = \sigma\left(g_h\left(g_c\left(g_E^{\text{user}}(u), g_E^{\text{item}}(i) \right) \right) \right), \tag{3}$$

where $\sigma(\cdot)$ is the sigmoid function. $g_E^{\text{user}}(\cdot)$ and $g_E^{\text{item}}(\cdot)$ embed the index of each user and item into a shared latent space. Notice that if we replace the index with some raw features of users and items, the framework will be able to predict for unseen users and items. The embeddings $\boldsymbol{v}_u = g_E^{\text{user}}(u)$ and $\boldsymbol{v}_i = g_E^{\text{user}}(i)$ would adjust to the empirical preference distribution during the training period. $g_c(\cdot)$ is the concatenation operation. The hidden layer g_h adopts the multi-layer structure as $g_h(\cdot) = g_{x_R}(\cdots g_1(\cdot)\cdots)$.

Most generative models directly learn the rating scores in the regression manner. However, two factors determine that the value itself is not so essential: (1) diverse rating standards are adopted by different users. The same rating represents different preference, which is known as the ***calibration drawback*** or ***multi-criteria phenomenon*** [2]; (2) a fixed score interval with discrete values is oversimplified, and thus a large amount of preference information would be ignored. Therefore, collaborative ranking methods are proposed to evaluate the ranking of items for each user, rather than predict the rating score for each item. One effective way to deal with the problem is to estimate the score difference between a pair of items and preserve the order. Specifically, two generators $g_\theta(u,i)$ and $g_\theta(u,j)$ are equipped in our framework:

$$\Delta G_{\theta,t} = g_\theta(u,i) - g_\theta(u,j),\ \forall\ t = (u,i,j) \in \mathcal{T}. \tag{4}$$

Discriminator. The traditional discriminative models for collaborative ranking merely learn a classifier to predict the label of each comparison. Such a paradigm requires sufficient training data to obtain a sophisticated classifier. In reality, the supervision signals for recommendation task are always in shortage [38]. Therefore, we design a new type of discriminative model to better force the score difference of each pair to approach the pursued margin. We construct two types of instances for training the discriminator. For each instance, it includes the code of a comparison like $t = (u,i,j)$ and the difference of scores between item i and j. If the discrepancy of scores comes from the generator (4), we give the corresponding instance a label -1. An instance will be labeled as 1 if it equips with an ideal score disparity, which is the maximum of all possible values and the pursued margin in classification. For each comparison $t = (u,i,j) \in \mathcal{T}$ and the two instances with different labels, the discriminator would be invalid if the generated discrepancy of scores gets close to the hypothetical one. Here we train logistic regression to maximize the probability of assigning the correct label to both desired training examples and samples generated from g, for each $t = (u,i,j) \in \mathcal{T}$ and the score discrepancy ΔG_t,

$$d_\phi\left(\Delta G_t, \boldsymbol{c}_{uij}, z_{uij}\right) = \sigma\left(z_{uij} \cdot f_h\left(f_c\left(\Delta G_t, \boldsymbol{c}_{uij}\right)\right)\right) \tag{5}$$

where $\boldsymbol{c}_{uij}$ denotes the code of triplet (u,i,j), z_{uij} is the indicator of ΔG_t

$$\Delta G_t = \begin{cases} \Delta G_{\theta,t} & \text{if } z_{uij} = -1, \\ \Delta G_t^* & \text{if } z_{uij} = \ \ 1, \end{cases} \tag{6}$$

and ΔG_t^* is the ideal case; f_c is the concatenation layer; the hidden layers are represented as $f_h(\cdot) = f_{x_D}(\cdots f_1(\cdot)\cdots)$. The probability can be further abbreviated as $d_\phi(\Delta G_t)$ in the following discussion.

Overall objective function. In summary, we let the two agents, the generator and the discriminator, play a minimax game: the generator tries to minimize the gap between the difference of generated scores and the pursued margin. Meanwhile, the discriminator tries to distinguish the difference between them. Finally, based on the learned scores, the ranking of items could be consistent with the preference distribution. The overall objective function can be formulated as:

$$\begin{aligned} \min_\theta \max_\phi J(G,d) = \mathbb{E}_{\Delta G_t^* \sim \mathcal{P}_{\Delta G^*}} \left[\log d_\phi\left(\Delta G_t^*\right)\right] + \\ \mathbb{E}_{t \sim \mathcal{P}_\mathcal{T}} \left[\log\left(1 - d_\phi\left(\Delta G_{\theta,t}\right)\right)\right], \end{aligned} \tag{7}$$

where $t = (u,i,j)$ is an observed triplet and $t \in \mathcal{T}$, $\mathcal{P}_\mathcal{T}$ is the empirical preference distribution, and $\mathcal{P}_{\Delta G^*}$ is the pursued score difference distribution. To generate a well-matched score function and improve the overall performance, we optimize the objective function (7) in an alternative optimization framework. The details are discussed in the following part.

Algorithm 1 Minimax Game for CRGAN

Input: generator G; discriminator D; training pairwise comparisons set $\mathcal{T}$.

```
Generate codes of each triple (u, i, j) in 𝒯 for D.
Initialize G, D with random parameters θ, ϕ.
for number of training epochs do
    for D-steps do
        The generator generates score difference for each triple
        (u, i, j) in 𝒯
        Update the discriminator via Eq. (8)
    end for
    for G-steps do
        Update the generator via Eq. (9)
    end for
end for
```

3.3 Optimization

Fix g, update d. The discriminator aims to discriminate the generated difference of the learned score from the pursued margin. Given the set of observed comparisons $\mathcal{T}$ and score difference generated from the current optimal generator $g_{\tilde{\theta}}$, the parameters of the discriminator then can be updated as follows:

$$\begin{aligned} \phi^* = \arg\max_\phi \mathbb{E}_{\Delta G_t^* \sim \mathcal{P}_{\Delta G^*}} \left[\log d_\phi\left(\Delta G_t^*\right)\right] + \\ \mathbb{E}_{t \sim \mathcal{P}_\mathcal{T}} \left[\log\left(1 - d_\phi\left(\Delta G_{\tilde{\theta},t}\right)\right)\right] \end{aligned} \tag{8}$$

Fix d, update g. Acting as the discriminator's adversary, the generator fits the observed preference distribution and increases the score difference between a pair of items, which would fool the discriminator. Specifically, if the discriminator has been fixed with the parameter ϕ^*, the objective function can be denoted as follows:

$$\begin{aligned} \tilde{\theta} &= \arg\min_\theta \mathbb{E}_{\Delta G_t^* \sim \mathcal{P}_{\Delta G^*}} \left[\log d_{\phi^*}\left(\Delta G_t^*\right)\right] + \\ &\qquad \mathbb{E}_{t \sim \mathcal{P}_\mathcal{T}} \left[\log\left(1 - d_{\phi^*}\left(\Delta G_{\theta,t}\right)\right)\right] \\ &= \arg\min_\theta \mathbb{E}_{t \sim \mathcal{P}_\mathcal{T}} \left[\log\left(1 - d_{\phi^*}\left(\Delta G_{\theta,t}\right)\right)\right] \\ &= \arg\max_\theta \mathbb{E}_{t \sim \mathcal{P}_\mathcal{T}} \left[\log\left(d_{\phi^*}\left(\Delta G_{\theta,t}\right)\right)\right] \end{aligned} \tag{9}$$

where d_{ϕ^*} is the fixed discriminator.

We adopt different user-item representations in the generator and discriminator. The embedding vectors for the generator will be updated, while the codes for the discriminator are generated with the off-shelf hashing methods off-line and all fixed. On one hand, the generator focuses on the score function; adaptive embedding vectors are needed to learn more reasonable representations of users and items. On the other hand, the discriminator focuses on classification, and fixed codes are enough to provide personalized discrimination. This feature enjoys two advantages: (1) If the non-differentiable embedding layer acts as a middle layer of the whole adversarial framework, back-propagation algorithm will be invalid when trying to optimize the generator. Therefore, adopting pre-generated and fixed encoding vectors as the input to the discriminator makes it convenient to apply traditional gradient descent algorithm. Although this problem can also be solved by policy gradient descent based on reinforcement learning, it becomes more time-consuming and less efficient. (2) There is an observation in GAN that the loss of the discriminative model converges quickly to zero during training. Then, no reliable path for gradient updating is provided with the generative model, and the game becomes invalid [1]. The generator in our framework simultaneously learns representation and non-linear score function, which can capture the characteristic of user preference and lead the discriminator to make more mistakes. Thus, the training process tends to be more efficient.

In summary, the overall solution of the entire adversarial framework is shown in Algorithm 1.

3.4 Advantages

Now we show that the traditional generative model of collaborative ranking is the special case of our generic framework.

In view of probability, the discriminator (5) learns the conditional probability distribution $\mathcal{P}\left(z_{uij}|\Delta G_t, y_{uij}, \phi\right)$. Then, the overall objective (7) has the following formulation:

$$\min_{\theta} \max_{\phi} J(G,d) = \sum_{(u,i,j)\in\mathcal{T}} \log\left(\mathcal{P}\left(z_{uij}=1|\Delta G_t^*, y_{uij}, \phi\right)\right) + \sum_{(u,i,j)\in\mathcal{T}} \log\left(\mathcal{P}\left(z_{uij}=-1|\Delta G_{\theta,t}, y_{uij}, \phi\right)\right). \tag{10}$$

When fixing d and updating g, the objective can be simplified as:

$$\min_{\theta} \sum_{(u,i,j)\in\mathcal{T}} \log\left(\mathcal{P}\left(z_{uij}=-1|\Delta G_{\theta,t}, y_{uij}; \phi^*\right)\right). \tag{11}$$

By Bayes' rule, the conditional probability distribution can be derived as:

$$\mathcal{P}(z_{uij}=-1|\Delta G_{\theta,t}, y_{uij}; \phi^*) = \frac{\mathcal{P}(\Delta G_{\theta,t}, y_{uij}|z_{uij}=-1;\phi^*)\mathcal{P}(z_{uij}=-1;\phi^*)}{\mathcal{P}(\Delta G_{\theta,t}, y_{uij};\phi^*)}. \tag{12}$$

where the marginal distribution $\mathcal{P}(\Delta G_{\theta,t}, y_{uij};\phi^*)$ and $\mathcal{P}(z_{uij} = -1;\phi^*)$ are both constants. Therefore, the function (11) can be derived as:

$$\min_{\theta} \sum_{(u,i,j)\in\mathcal{T}} \log\left(\mathcal{P}\left(\Delta G_{\theta,t}, y_{uij}|z_{uij}=-1; \phi^*\right)\right). \tag{13}$$

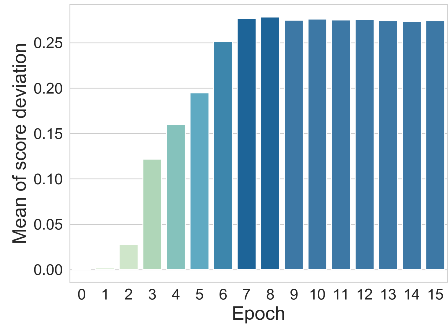

Figure 2: Mean value of score difference w.r.t. the number of epochs on MovieLens100K. The discriminator forces the score difference of pairs to approach the pursued margin.

Table 1: Details of the datasets

	MoiveLens100K	MoiveLens1M	Netflix
#User	943	6,040	10,000
#Item	1,683	3,706	17,770
#Rating	100,000	1,000,209	8,930,336
Sparsity	93.70%	95.53%	94.97%
#Pair	426,010	3,334,252	7,916,812
#Pair / #user	451	552	792
#Train item / #item	29.59%	21.31%	5.40%
#Negative sample	54,383	687,774	563,067

If we let $\mathcal{L}'_{\mathrm{g}}(\cdot) = \log(\cdot)$ and $\sigma'(\cdot) = d_{\phi^*}(\cdot)$, the function (13) can be represented as:

$$\min_{\theta} \sum_{(u,i,j)\in\mathcal{T}} \mathcal{L}'_{\mathrm{g}}\left(\mathcal{P}\left(\sigma'\left(\Delta G_{\theta,t}\right), y_{uij}\right)\right). \tag{14}$$

Intuitively, if we adopt a predefined score function and do not update the discriminator, CRGAN will degenerate to a traditional generative method, which reveals why our framework could achieve superior performance: For one thing, the adaptive score function is learned for specific preference by the deep generator. For another, the discriminator provides personalized supervision signals for the generator so that the score differences continually approximate the ideal margin.

To illustrate such learning process of the generator, we visualize the change of the mean values of score difference *w.r.t* epochs in Figure 2. The generator is randomly initialized so that the mean of score difference almost equals to 0 in the first few epochs. Then, we can see that the score difference gradually increases as the gaming goes on, which is in line with our expectation.

4 EXPERIMENT

We perform evaluations on three popular benchmark datasets: MoiveLens-100K[11], MovieLens1M[11] and Netflix[28]. The details are summarized in Table 1.

4.1 MovieLens100K

Dataset description. The full MovieLens dataset contains 26,000,000 integer movie ratings ranging from 1 to 5 collected from the MovieLens website. We first use its subset MovieLens100K (*i.e.*, including

100,000 ratings) to generate pairwise data and evaluate the collaborative ranking algorithms.

Following the experiment setting of [31], we randomly select $N = 50$ ratings for each user to generate pairwise preference for training by comparing the ratings for each pair of items. The item pairs with equal ratings are ignored. The rest ratings form the test set. Note that those items not appearing in the training set are filtered out.

Evaluation Metrics. To evaluate the performance of the algorithm, we adopt six standard ranking performance measures [2, 9, 13, 42]: Hit Rate (HR@K), Precision (P@K), Normalized Discounted Cumulative Gain (NDCG@K), Area Under the ROC Curve (AUC@K), Mean Average Precision (MAP@K) and Mean Reciprocal Rank (MRR@K). *@K* means that in the test set, Top-K items are positive instances, and others are negative ones.

Competitors. To investigate the performance of our proposed method, we implement the following competitors:

- IRGAN [42]: This is a pioneering model applying adversarial learning to solve collaborative filtering problems.
- MLP [13]: This deep generative model adopts a multilayer perceptron to learn a non-linear score function.
- GMF [13]: This instantiation of NCF applies a linear kernel to learn score function and can be regarded as a generalized MF method.
- NeuMF [13]: This model fuses the pre-trained MLP and GMF, and separate embeddings are learned to better model a more complex score function.
- CofiRank [45]: This MF method directly optimizes ranking metrics such as NDCG.
- RegularizedSVD [32]: This classic MF model optimizes squared error with regularization for both users and items.
- RankBasedSVD [19]: This traditional MF method aims at minimizing various ranking loss function.
- Local Collaborative Ranking (LCR) [19]: Under the assumption that the rating matrix is locally low-rank, LCR combines multiple models by weighted sum.
- Primal-CR [46]: With the utilization of Newton's method and a well-designed calculation of the gradient and Hessian matrix, this discriminative method has made amazing progress in reducing time complexity.
- Primal-CR++ [46]: This method improves the performance of Primal-CR by further accelerating the gradient and Hessian calculation.
- AltSVM [31]: It provides a non-convex algorithm alternatively optimizing user and item latent vector U and V by stochastic dual coordinate descent algorithm. To our best knowledge, this is one of the state-of-the-art discriminative CR model.
- Global Ranking [31]: This model removes the personalization in AltSVM, namely, the problem is solved by only optimizing the latent vector V.

Implementation details. We implement CRGAN using the popular Keras package [8]. The learnable embedding layers are initialized with the normal distribution $\mathcal{N}(0, 1)$. Similar to [13], we set the number of hidden neurons to $128(Relu) \rightarrow 64(Relu) \rightarrow 32(Relu) \rightarrow 8(Relu)$ for FC layers in the generator. Then the discriminator is built with a simpler architecture: $32(Relu) \rightarrow 8(Relu)$ to ensure a better game process. For a faster convergence, we use Adam [17] as the full-batch gradient descent optimizer and set the learning rate to 0.05 and 0.01 for the generator and discriminator, respectively. We set a large #epochs and adopt early stopping strategy to stop training when the loss becomes stable. Besides, we implement a variant of our CRGAN where the embedding vectors for the generator are fixed (denoted as *CRGAN-fixed*) to demonstrate the effectiveness of dynamic representations.

Table 2: Experimental results of MovieLens100K. For each metric, the highest value is marked with red color and the second highest is marked with green.

	HR@10↑	P@10↑	NDCG@10↑	AUC@10↑	MAP@10↑	MRR@10↑
IRGAN [42]	0.8511	0.2642	0.2670	0.5989	0.2858	0.1020
MLP [13]	0.9598	0.3519	0.3909	0.7271	0.3938	0.1491
GMF [13]	0.9175	0.3437	0.3580	0.7287	0.3762	0.1331
NeuMF [13]	0.9416	0.3590	0.3757	0.7340	0.3858	0.1382
CoFiRank [45]	0.9256	0.3354	0.3598	0.7152	0.3717	0.1357
RegularizedSVD [32]	0.8934	0.3404	0.3505	0.7287	0.3702	0.1298
RankBasedSVD [19]	0.9256	0.3318	0.3510	0.7042	0.3639	0.1318
LCR [19]	0.9437	0.3503	0.3744	0.7245	0.3830	0.1404
Primal-CR [46]	0.9577	0.3899	0.4205	0.7160	0.4128	0.1539
Primal-CR++ [46]	0.9575	0.3896	0.4199	0.7165	0.4139	0.1526
AltSVM [31]	0.9618	**0.3899**	**0.4209**	0.7186	**0.4136**	**0.1542**
Global Ranking [31]	**0.9678**	0.3795	0.4045	**0.7353**	0.4083	0.1483
CRGAN-fixed	0.9396	0.3455	0.3790	0.6555	0.3661	0.1412
CRGAN(ours)	**0.9839**	**0.4678**	**0.4864**	**0.7559**	**0.4753**	**0.1688**

Results. As shown in Table 2, we can see that our method significantly outperforms all the other models on all the metrics. It is worth mentioning that CRGAN outperforms the second best model by 7.79%, 6.55%, 6.17% in terms of P, NDCG and MAP, respectively. From the results, we can make the following observations: 1) The performance improvement of CRGAN and MLP over models like RegularizedSVD, RankBasedSVD and LCR shows the importance of non-linear transformations. 2) Furthermore, CRGAN outperforms all the three implementations of NCF, confirming that the introduction of adversarial learning provides a more powerful discrimination ability for our model. 3) However, IRGAN obtains a much lower performance on this dataset. This may be caused by that IRGAN ignores the sample's preference order, which puts it at a serious disadvantage in top-K recommendation task. 4) Dynamic representations help to learn a better generator, as the dynamic representations beat the fixed ones with a clear margin. Overall, such results verify the effectiveness of our proposed framework.

4.2 MovieLens1M

Dataset description. This dataset is a larger subset of the full MovieLens dataset, containing 1,000,209 ratings. It has a larger scale and is also more sparse than MovieLens100K. In MoiveLens1M, only 3839 of the 6040 users have rated more than 60 items.

Implementation details. All the hyperparameters are the same as MoiveLens100K.

Results. The performance of all the involved models is recorded in Table 2. The corresponding results show that CRGAN consistently outperforms the competitors on all the metrics, which is the same on MovieLens100K. In particular, the performance gap between CRGAN and the second best model is 8.47%, 7.12% and 6.14% in terms of P, NDCG and MAP, respectively. Therefore, the effectiveness of our framework is again validated.

Table 3: Experimental results of MovieLens1M. For each metric, the highest value is marked with red color and the second highest is marked with green.

	HR@10↑	P@10↑	NDCG@10↑	AUC@10↑	MAP@10↑	MRR@10↑
IRGAN [42]	0.9309	0.3425	0.3632	0.7290	0.3623	0.1341
MLP [13]	0.9439	0.3489	0.3832	0.7353	0.3763	0.1431
GMF [13]	0.9342	0.3503	0.3837	0.7327	0.3770	0.1432
NeuMF [13]	0.9469	0.3543	0.3964	0.7374	0.3863	0.1492
CoFiRank [45]	0.8677	0.2799	0.3054	0.6553	0.3060	0.1176
RegularizedSVD [32]	0.8946	0.3333	0.3588	0.7371	0.3624	0.1343
RankBasedSVD [19]	0.9530	0.3510	0.3945	0.7324	0.3855	0.1494
LCR [19]	0.9543	0.3524	0.3953	0.7280	0.3847	0.1491
Primal-CR [46]	0.9030	0.3289	0.3512	0.7048	0.3520	0.1318
Primal-CR++ [46]	0.9027	0.3295	0.3507	0.7050	0.3515	0.1314
AltSVM [31]	0.9518	0.3641	0.3972	0.7393	0.3860	0.1468
Global Ranking [31]	0.9523	0.3549	0.3985	0.7390	0.3895	0.1505
CRGAN(ours)	0.9820	0.4488	0.4697	0.7613	0.4509	0.1632

Table 4: Experimental results of Netflix. For each metric, the highest value is marked with red color and the second highest is marked with green.

	HR@10↑	P@10↑	NDCG@10↑	AUC@10↑	MAP@10↑	MRR@10↑
IRGAN [42]	0.9575	0.2859	0.3106	0.6374	0.3210	0.1244
MLP [13]	0.9673	0.2982	0.3195	0.6683	0.3346	0.1270
GMF [13]	0.9465	0.2603	0.2756	0.6223	0.2956	0.1123
NeuMF [13]	0.9642	0.2884	0.3060	0.6598	0.3234	0.1221
CoFiRank [45]	0.9499	0.2604	0.2737	0.6310	0.2964	0.1118
RegularizedSVD [32]	0.9754	0.3185	0.3406	0.6937	0.3548	0.1335
RankBasedSVD [19]	0.9510	0.2764	0.3033	0.6455	0.3198	0.1237
LCR [19]	0.9549	0.2781	0.3066	0.6310	0.3168	0.1243
Primal-CR [46]	0.9475	0.2798	0.3015	0.6472	0.3213	0.1219
Primal-CR++ [46]	0.9471	0.2797	0.3015	0.6471	0.3213	0.1219
AltSVM [31]	0.9654	0.3041	0.3292	0.6781	0.3452	0.1311
Global Ranking [31]	0.9690	0.2998	0.3249	0.6732	0.3404	0.1298
CRGAN(ours)	0.9890	0.3833	0.4100	0.6953	0.4011	0.1511

4.3 Netflix

Dataset description. This dataset comes from the famous open competition Netflix Prize. It includes 100,480,507 ratings for 17,770 movies collected from 480,189 users without any additional information like actors or summary provided. Since this dataset is much larger than MovieLens datasets, we randomly sample 10,000 users with their rated movies for the evaluation.

Implementation details. Considering the scale of the dataset, the architectures of the generator and discriminator are condensed as $128(Relu) \rightarrow 32(Relu) \rightarrow 8(Relu)$ and $16(Relu) \rightarrow 4(Relu)$, respectively. The learning rates are the same as MoiveLens100K. For a fair comparison, for users with more than M ratings, we randomly sample M items for testing. To verify the performance of our model on different-sized test sets, M ranges from 70 to 200.

Results. We record the results for $M = 70$ in Table 4 and also plot the metrics as M increases in Figure 3. We can observe that our model consistently achieves the highest performance with varying test data sizes. Particularly, the performance gap becomes larger as the size of test data grows up. Besides, it is interesting to note that, different from other metrics, AUC increases with respect to M. The reason is that the number of positive samples is fixed to 10, while negative ones are increasing as M becomes larger, resulting in more and more negative samples being ranked higher than positive ones.

After the quantitative comparison, we then visualize the predictions to further investigate the performance. First, we visualize the top-10 prediction results of CRGAN, AltSVM and RegularizedSVD for $M = 70$ in Figure 4. For each heatmap, the horizontal axis represents 100 randomly selected users, and the vertical axis denotes the top-10 test items of each user from top to bottom. The grid with a deeper color represents a higher prediction score, while the lightest color indicates that the item is not included in the top-10 predicted list. From these three heatmaps, it is obvious to see that our model obtains deeper grids, and they tend to hit the top half of the figure, leading to the better performance in terms of both accuracy and ranking.

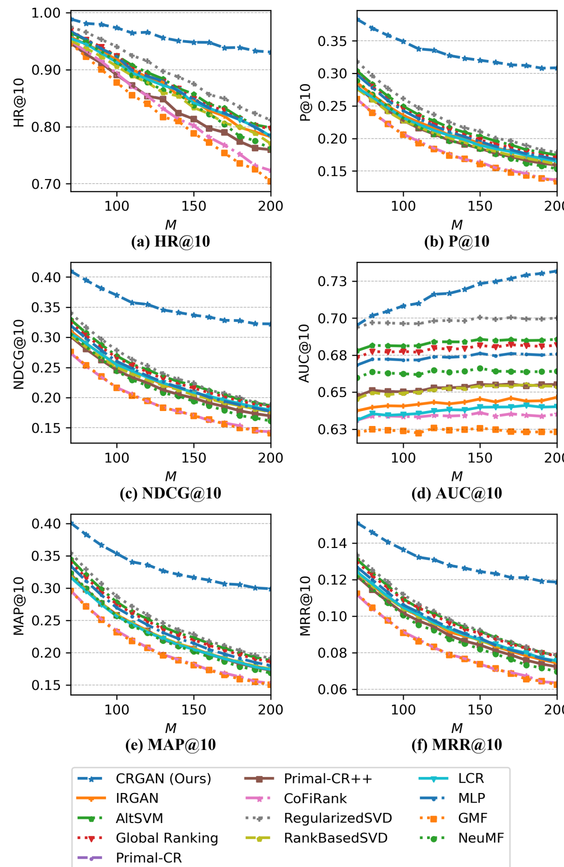

Figure 3: Performance comparisons on Netflix w.r.t. the maximum of test items for each user. CRGAN outperforms competitors with varying test data sizes.

Then we calculate the normalized rating for all items, which is simply equivalent to averaging the heatmap in Figure 4 along the horizontal axis. We plot all the normalized ratings of CRGAN and RegularizedSVD for $M = 70$ in Figure 5. The ideal curve is monotonically decreasing. We can see that the curve of CRGAN is much more smooth than RegularizedSVD, especially in the positive sample area. Moreover, there are less abnormal points that locate in negative sample area for CRGAN, which are predicted higher than some positive samples. Therefore, CRGAN obtains a higher AUC score than RegularizedSVD.

Do the generator and discriminator really play an effective game? We plot the change of metrics in Figure 6 to answer this question.

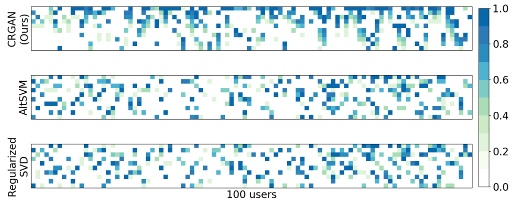

Figure 4: Visualization of ranking results for the top-10 test items on Netflix. The grid with a deeper color represents a higher prediction score. Obviously, our method obtains more grids with deep colors, and they tend to hit the top half of the figure.

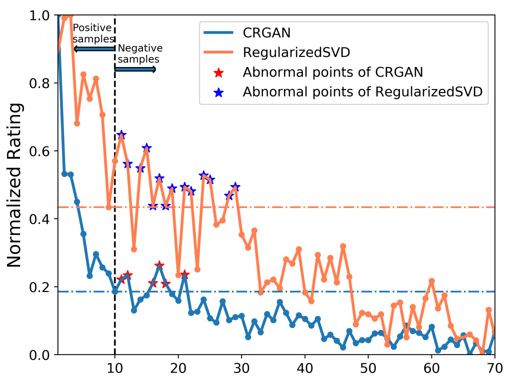

Figure 5: Normalized rating for test items on Netflix. The ideal rating is monotonically decreasing w.r.t. the true ranking. The curve of CRGAN is much more smooth than RegularizedSVD, especially in the positive sample area.

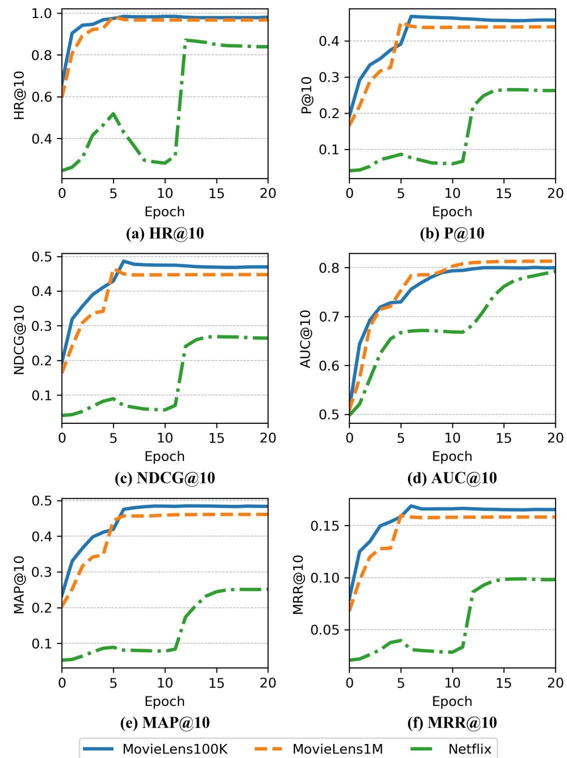

Figure 6: All the metrics of CRGAN w.r.t. the number of epochs on three datasets. The ups and downs of the curves show the process of effective adversarial learning.

In smaller datasets such as MovieLens100K and MovieLens1M, the gaming process is not very distinct, but it can be still observed that the metrics are increasing by stages. Also, the increasing trend of metrics is consistent with that of the score difference in Figure 2, which shows the essential role of the ranking margin. In a larger dataset like Netflix, user preference is more complicated, resulting in an intense game process: 1) In the first few epochs, discriminator is much stronger than the generator. Consequently, most metrics increase quite slowly. However, AUC enjoys a faster increasing because the margins between the positive and negative samples expand more quickly than those between positive samples. 2) After enough training, the generator gradually narrows the gap between learned difference and pursued margin. 3) Finally, the generated score differences are so consistent with the preference distribution that the discriminator cannot distinguish them from the pursued margin. As a result, all the metrics rapidly increase to a high level. In summary, our framework benefits from an effective adversarial game.

5 CONCLUSION

In this paper, we propose a novel generative adversarial framework called CRGAN to solve the collaborative ranking problem under the pairwise comparison setting. In this framework, we approximate an arbitrary continuous score function by a deep generator, which shows adaptability to specific datasets. Meanwhile, to better estimate the ranking of items, a discriminator provides personalized supervision signals to force the score difference to approach the pursued margin. When the game reaches equilibrium, the rankings of generated scores are consistent with the preference distribution, and thus the ranking performance will be improved. The increasing trends of the score difference and accuracy metrics are basically synchronized, which is in line with our expectation. Moreover, we adopt different user-item representations in the generator and discriminator so that the two agents can be effectively updated by gradient descent. Further analysis shows that our framework is generic and can generalize the traditional generative model. In our empirical studies, we perform a series of experiments on three real-world datasets: MovieLens100K, MovieLens1M and Netflix. The corresponding results show the superiority of our proposed model.

ACKNOWLEDGMENTS

This work was supported in part by National Natural Science Foundation of China (No. 61620106009, U1636214, 61733007, U1736219, 61672514), in part by Key Research Program of Frontier Sciences, CAS: QYZDJ-SSW-SYS013, in part by Beijing Natural Science Foundation (No. L182057, 4182079), and in part by Youth Innovation Promotion Association CAS.

REFERENCES

[1] Martín Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein Generative Adversarial Networks. In *International Conference on Machine Learning*. 214–223.

[2] Suhrid Balakrishnan and Sumit Chopra. 2012. Collaborative ranking. In *International Conference on Web Search and Data Mining*. 143–152.

[3] Shlomo Berkovsky, Ronnie Taib, and Dan Conway. 2017. How to Recommend?: User Trust Factors in Movie Recommender Systems. In *International Conference on Intelligent User Interfaces*. 287–300.

[4] Lucas Bernardi, Jaap Kamps, Julia Kiseleva, and Melanie J. I. Müller. 2015. The Continuous Cold-start Problem in e-Commerce Recommender Systems. In *ACM Conference on Recommender Systems*. 30–33.

[5] Xiaoyan Cai, Junwei Han, and Libin Yang. 2018. Generative Adversarial Network Based Heterogeneous Bibliographic Network Representation for Personalized Citation Recommendation. In *AAAI Conference on Artificial Intelligence*. 5747–5754.

[6] Dong-Kyu Chae, Jin-Soo Kang, Sang-Wook Kim, and Jung-Tae Lee. 2018. CFGAN: A Generic Collaborative Filtering Framework based on Generative Adversarial Networks. In *ACM International Conference on Information and Knowledge Management*. 137–146.

[7] Yu Chen, Chunhua Shen, Xiu-Shen Wei, Lingqiao Liu, and Jian Yang. 2017. Adversarial PoseNet: A Structure-Aware Convolutional Network for Human Pose Estimation. In *IEEE International Conference on Computer Vision*. 1221–1230.

[8] François Chollet et al. 2015. Keras. https://github.com/fchollet/keras.

[9] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of recommender algorithms on top-n recommendation tasks. In *ACM Conference on Recommender Systems*. 39–46.

[10] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*. 2672–2680.

[11] F. Maxwell Harper and Joseph A. Konstan. 2016. The MovieLens Datasets: History and Context. *ACM Transactions on Internet and Information Systems* 5, 4 (2016), 19:1–19:19.

[12] Bin He, Feng Gao, Daiqian Ma, Boxin Shi, and Ling-Yu Duan. 2018. ChipGAN: A Generative Adversarial Network for Chinese Ink Wash Painting Style Transfer. In *ACM International Conference on Multimedia*. 1172–1180.

[13] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *International Conference on World Wide Web*. 173–182.

[14] Shintami Chusnul Hidayati, Cheng-Chun Hsu, Yu-Ting Chang, Kai-Lung Hua, Jianlong Fu, and Wen-Huang Cheng. 2018. What Dress Fits Me Best?: Fashion Recommendation on the Clothing Style for Personal Body Shape. In *ACM International Conference on Multimedia*. 438–446.

[15] Cho-Jui Hsieh, Nagarajan Natarajan, and Inderjit S. Dhillon. 2015. PU Learning for Matrix Completion. In *International Conference on Machine Learning*. 2445–2453.

[16] Jun Hu and Ping Li. 2017. Decoupled Collaborative Ranking. In *International Conference on World Wide Web*. 1321–1329.

[17] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*.

[18] Oluwasanmi Koyejo, Sreangsu Acharyya, and Joydeep Ghosh. 2013. Retargeted matrix factorization for collaborative filtering. In *ACM Conference on Recommender Systems*. 49–56.

[19] Joonseok Lee, Samy Bengio, Seungyeon Kim, Guy Lebanon, and Yoram Singer. 2014. Local Collaborative Ranking. In *International Conference on World Wide Web*. 85–96.

[20] Yaoyu Li, Tianzhu Zhang, Lingyu Duan, and Changsheng Xu. 2018. A Unified Generative Adversarial Framework for Image Generation and Person Reidentification. In *ACM International Conference on Multimedia*. 163–172.

[21] Chu-Hsing Lin and Hsuan Chi. 2019. A Novel Movie Recommendation System Based on Collaborative Filtering and Neural Networks. In *Advanced Information Networking and Applications*. 895–903.

[22] Kevin Lin, Dianqi Li, Xiaodong He, Ming-Ting Sun, and Zhengyou Zhang. 2017. Adversarial Ranking for Language Generation. In *Advances in Neural Information Processing Systems*. 3158–3168.

[23] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon.com Recommendations: Item-to-Item Collaborative Filtering. *IEEE Internet Computing* 7, 1 (2003), 76–80.

[24] Tie-Yan Liu. 2009. Learning to Rank for Information Retrieval. *Foundations and Trends in Information Retrieval* 3, 3 (2009), 225–331.

[25] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhen Wang, and Stephen Paul Smolley. 2017. Least Squares Generative Adversarial Networks. In *IEEE International Conference on Computer Vision*. 2813–2821.

[26] Mehdi Mirza and Simon Osindero. 2014. Conditional Generative Adversarial Nets. *CoRR* abs/1411.1784 (2014). arXiv:1411.1784 http://arxiv.org/abs/1411.1784

[27] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529–533.

[28] Netflix. 2009. Netflix Prize Data Set. (2009).

[29] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. 2016. f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization. In *Advances in Neural Information Processing Systems*. 271–279.

[30] Weike Pan and Li Chen. 2013. GBPR: Group Preference Based Bayesian Personalized Ranking for One-Class Collaborative Filtering. In *International Joint Conference on Artificial Intelligence*. 2691–2697.

[31] Dohyung Park, Joe Neeman, Jin Zhang, Sujay Sanghavi, and Inderjit S. Dhillon. 2015. Preference Completion: Large-scale Collaborative Ranking from Pairwise Comparisons. In *International Conference on Machine Learning*. 1907–1916.

[32] Arkadiusz Paterek. 2007. Improving regularized singular value decomposition for collaborative filtering. In *ACM International Conference on Knowledge Discovery and Data Mining*. 5–8.

[33] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Conference on Uncertainty in Artificial Intelligence*. 452–461.

[34] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2011. Introduction to Recommender Systems Handbook. In *Recommender Systems Handbook*. 1–35.

[35] Yue Shi, Alexandros Karatzoglou, Linas Baltrunas, Martha Larson, Nuria Oliver, and Alan Hanjalic. 2013. CLiMF: Collaborative Less-Is-More Filtering. In *International Joint Conference on Artificial Intelligence*. 3077–3081.

[36] Si Si, Kai-Yang Chiang, Cho-Jui Hsieh, Nikhil Rao, and Inderjit S. Dhillon. 2016. Goal-Directed Inductive Matrix Completion. In *ACM International Conference on Knowledge Discovery and Data Mining*. 1165–1174.

[37] Nathan Srebro, Jason D. M. Rennie, and Tommi S. Jaakkola. 2004. Maximum-Margin Matrix Factorization. In *Advances in Neural Information Processing Systems*. 1329–1336.

[38] Xiaoyuan Su and Taghi M. Khoshgoftaar. 2009. A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence* 2009 (2009), 421425:1–421425:19.

[39] Sandeep Subramanian, Sai Rajeswar, Francis Dutil, Chris Pal, and Aaron C. Courville. 2017. Adversarial Generation of Natural Language. In *Representation Learning for NLP*. 241–251.

[40] Richard S. Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. 1999. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *Advances in Neural Information Processing Systems*. 1057–1063.

[41] Maksims Volkovs and Richard S. Zemel. 2012. Collaborative Ranking With 17 Parameters. In *Advances in Neural Information Processing Systems*. 2303–2311.

[42] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. IRGAN: A Minimax Game for Unifying Generative and Discriminative Information Retrieval Models. In *International ACM Conference on Research and Development in Information Retrieval*. 515–524.

[43] Qinyong Wang, Hongzhi Yin, Zhiting Hu, Defu Lian, Hao Wang, and Zi Huang. 2018. Neural Memory Streaming Recommender Networks with Adversarial Training. In *ACM International Conference on Knowledge Discovery and Data Mining*. 2467–2475.

[44] Xin Wang, Steven C. H. Hoi, Martin Ester, Jiajun Bu, and Chun Chen. 2017. Learning Personalized Preference of Strong and Weak Ties for Social Recommendation. In *International Conference on World Wide Web*. 1601–1610.

[45] Markus Weimer, Alexandros Karatzoglou, Quoc V. Le, and Alexander J. Smola. 2007. COFI RANK - Maximum Margin Matrix Factorization for Collaborative Ranking. In *Advances in Neural Information Processing Systems*. 1593–1600.

[46] Liwei Wu, Cho-Jui Hsieh, and James Sharpnack. 2017. Large-scale Collaborative Ranking in Near-Linear Time. In *ACM International Conference on Knowledge Discovery and Data Mining*. 515–524.

[47] Jingwei Xu, Yuan Yao, Hanghang Tong, Xianping Tao, and Jian Lu. 2017. HoORaYs: High-order Optimization of Rating Distance for Recommender Systems. In *ACM International Conference on Knowledge Discovery and Data Mining*. 525–534.

[48] Qianqian Xu, Jiechao Xiong, Xinwei Sun, Zhiyong Yang, Xiaochun Cao, Qingming Huang, and Yuan Yao. 2018. A Margin-based MLE for Crowdsourced Partial Ranking. In *ACM Multimedia Conference on Multimedia Conference*. 591–599.

[49] Jinfeng Yi, Rong Jin, Shaili Jain, and Anil K. Jain. 2013. Inferring Users' Preferences from Crowdsourced Pairwise Comparisons: A Matrix Completion Approach. In *AAAI Conference on Human Computation and Crowdsourcing*. 207–215.

[50] Zhou Zhao, Hanqing Lu, Deng Cai, Xiaofei He, and Yueting Zhuang. 2016. User Preference Learning for Online Social Recommendation. *IEEE Transactions on Knowledge and Data Engineering* 28, 9 (2016), 2522–2534.

[51] Meizi Zhou, Zhuoye Ding, Jiliang Tang, and Dawei Yin. 2018. Micro Behaviors: A New Perspective in E-commerce Recommender Systems. In *ACM International Conference on Web Search and Data Mining*. 727–735.

[52] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *IEEE International Conference on Computer Vision*. 2242–2251.