24th European Conference on Artificial Intelligence - ECAI 2020
Santiago de Compostela, Spain

Directional Adversarial Training for Recommender Systems

Yangjun Xu[1,2] and **Liang Chen**[1,2,*] and **Fenfang Xie**[1,2] and **Weibo Hu**[1,2]
Jieming Zhu [3] and **Chuan Chen**[1,2] and **Zibin Zheng**[1,2]

Abstract. Adversarial training is shown as an effective method to improve the generalization ability of deep learning models by making random perturbations in the input space during model training. A recent study has successfully applied adversarial training into recommender systems by perturbing the embeddings of users and items through a minimax game. However, this method ignores the collaborative signal in recommender systems and fails to capture the smoothness in data distribution. We argue that the collaborative signal, which reveals the behavioural similarity between users and items, is critical to modeling recommender systems. In this work, we develop the Directional Adversarial Training (DAT) strategy by explicitly injecting the collaborative signal into the perturbation process. That is, both users and items are perturbed towards their similar neighbours in the embedding space with proper restriction. To verify its effectiveness, we demonstrate the use of DAT on Generalized Matrix Factorization (GMF), one of the most representative collaborative filtering methods. Our experimental results on three public datasets show that our method (called DAGMF) achieves a significant accuracy improvement over GMF and meanwhile, it is less prone to overfitting than GMF.

1 Introduction

Adversarial training is a novel method proposed by [12] to address the problem that deep learning models can be easily fooled by adversarial examples [26], which are constructed by imposing small perturbation on the input examples. Most recently, several studies [12, 22, 23] have pointed out that adversarial training can work as a regularization method to improve the generalization performance as well as to prevent deep learning models from overfitting.

Although adversarial training achieves great success in the area of Computer Vision (CV), it is difficult to directly apply it to recommender systems. This is because the input data are discrete and mostly represented by high dimensional one-hot vectors (e.g, for users and items), which are different from the continuous values in the image domain. Directly adding noise to a discrete value is irrational since it will change the original semantics of input examples. Instead, a recent study, APR [14], extends adversarial training to recommender systems by adding perturbations in the embedding space towards the direction that can maximize the loss function. For simplicity, we refer to this perturbation strategy as maximum direction.

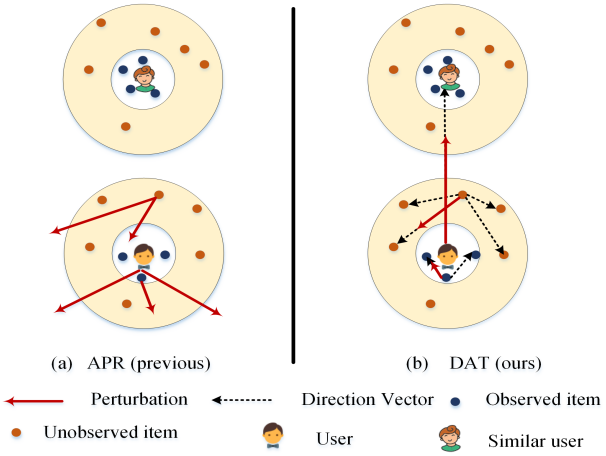


Figure 1. Intuitive sketch to explain the process of directional adversarial training. Observed items that have interaction with the user is in the small circle and other items in the large circle is unobserved items.

The basic strategy of adversarial training is that for an input example x, it will keep the same label for x and the adversarial example x_{adv} perturbed from x during training. This allows the model to consider more unseen space around x, thus encouraging the local smoothness among similar examples and further pushing the model towards better generalization [23]. However, as illustrated in Figure 1 (a), the maximum direction perturbation may move the embedding vector of example x close to the examples with different labels (dissimilar examples) or even non-existing examples. We present a detailed example through our preliminary experiments in Figure 2. Figure 2 (a) shows the original Top-10 nearest neighbors for the user with ID 16 and Figure 2 (b) shows the Top-10 nearest neighbors after adding maximum direction perturbation in the embedding space. As we can see, the Top-10 nearest neighbors have a large change under the maximum direction perturbation. It means that the current maximum direction strategy cannot keep the original semantic information hidden in the user-item interactions, and has not fully exploited the advantage of adversarial training for recommender systems.

In this paper, we tackle this problem by imposing proper restrictions on the perturbation direction, which we call Directional Adversarial Training (DAT). We perturb an example x towards another existing example x' in the embedding space and a weight w is set to control the distance x towards x'. DAT introduces an extra adversarial loss so that the training process plays a minimax game: the

[1] School of Data and Computer Science, Sun Yat-sen University, China, email: xyjlearner@gmail.com,{xieff5,huwb7}@mail2.sysu.edu.cn,{chenliang6,zhzibin,chenchuan,}@mail.sysu.edu.cn, *corresponding author
[2] National Engineering Research Center of Digital Life, Sun Yat-sen University, China.
[3] Huawei Noah's Ark Lab, China.

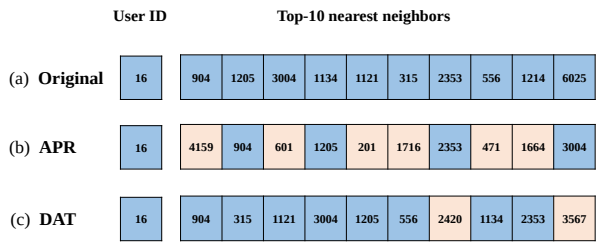


Figure 2. User Top-10 nearest neighbors. The number means the ID of the user. The blue colour denotes original nearest neighbors before adversarial training. The orange colour denotes new nearest neighbors after adding the perturbation.

weight w will be calculated via maximizing the Bayesian Personalized Ranking (BPR) loss [24]; next, the model is trained to minimize the BPR loss and DAT loss. Moreover, we consider yielding more effective embeddings by guiding the perturbation direction with the crucial collaborative signal in Collaborative Filtering (CF) recommender systems. The intuition is that users with similar behaviours would exhibit similar preference on items.

Figure 1 (b) illustrates our directional perturbation strategy. Given a user u, we make it perturb towards Top-K nearest neighbours in the embedding space based on pre-trained model parameters. Meanwhile, the perturbation direction of an observed item i and an unobserved item j will be perturbed towards the other observed items in the small circle and unobserved items in the large circle, respectively. In this way, the information of similar examples will flow between each other, which explicitly injects the collaborative signal into adversarial learning process. Figure 2 (c) shows the Top-10 nearest neighbors of user 16 after adding our direction perturbation, which is more coherent to the original.

The main contributions of this work are summarized as follows:

- We investigate the limitation of existing adversarial training methods in recommendation through our preliminary experiments.
- We propose a novel technique, Directional Adversarial Training (DAT), by restricting the direction of perturbation and explicitly encoding the collaborative signal into the adversarial learning process. We demonstrate the use of DAT on Generalized Matrix Factorization (GMF) [15], leading to our method called Directional Adversarial training Generalized Matrix Fa ctorization (DAGMF).
- We conduct experiments on three public datasets to verify the effectiveness of DAT for recommender systems. Specifically, DAGMF achieves significant improvements in both HR (HitRatio) and NDCG (Normalized Discounted Cumulative Gain) metrics compared with the state-of-the-art models.

2 Related Work

Recommendation. With the booming of information, it is a big challenge for users to find items that meet their preference. CF technique addresses this challenge by assuming that the users similar in behaviours show similar preference on items and focusing on exploiting user-item interactions. Neighbourhood-based CF [21] is one of the early and effective CF methods by utilizing explicit similar measurement (e.g., Euclidean distance, Cosine similarity) to calculate the interaction strength between users and items. Model-based CF is one of the most popular and widely used recommendation approaches in recent years. Matrix Factorization (MF) [19] approach is an important realization of model-based CF methods. It predicts unknown ratings based on the factorization of the original user-item rating matrix by mapping the one-hot vector of each user and item as an embedding vector and then conducting inner product between them. Whereas its linearity structure makes it fail to capture the complex and nonlinear interaction between users and items [15]. To this end, some recent works [15, 6] apply deep learning techniques to CF recommender systems, such as GMF. This is done by modelling user-item interactions with multi-layer perceptron (MLP), which can learn more powerful and expressive interaction function. Later on, to learn more effective embedding, much effort has been devoted to incorporating side information like text or image content feature [5], neighbour relations [29], attributes of users and items [20, 31] and collaborative signal [30, 7]. Moreover, the session-based recommendation [13] has been proposed to consider session-based data instead of CF method when modelling the user preference.

Adversarial Training. In the beginning, adversarial training is proposed to solve the issue that the state-of-the-art classification models are vulnerable to the adversarial examples due to their linearity structure [12]. It trains the model with adversarial examples which can be generated by the fast gradient sign method [12] effectively so as to improve the robustness of the model. Later on, Moreover, the idea of adversarial training has been extended to works on Natural Language Processing (NLP) tasks (e.g., sequence label) [22] and Network classifier tasks [11, 8], in which the perturbations are added on the embedding instead of the inputs. They demonstrate that their strategy can rather be seen as a regularizer to improve the generalization performance of the classifier model than work as a defence method.

It is worth noting that most of the existing works about adversarial training focus on CV domain. There are few studies to explore adversarial training for recommender systems. [28] proposes minimax game in recommender systems, namely IRGAN, based on the Generative Adversarial Nets (GANs) framework, which is to learn more effective embedding for MF. CFGAN [4] is the vector-wise GAN-based CF without learning the embedding vectors. They have very complicated frameworks and suffer from well-recognized hard training problems. APR [14] considers to exploiting the adversarial training in the recommender systems as a regularization with adding maximum perturbation in the embedding space ignoring the collaborative signal.

3 Preliminary

We first introduce the technical background for recommender systems and the formulation of GMF. Then, we recapitulate the Bayesian Personalized Ranking method, with a pairwise ranking loss function.

3.1 Problem Definition

Formally, we denote that there is a set of m users, $U = \{u_1, u_2, ..., u_m\}$, a set of n items, $I = \{i_1, i_1, ..., i_n\}$ and a sparse matrix $\mathbf{R}_{m \times n}$. An entry (u, i) in $\mathbf{R}$ denoted by r_{ui} is 1 if user u has interaction with item i and 0 otherwise. In the model-based recommender system, the input consists of two feature vectors v_u^U and v_i^I that describe user u and item i, respectively. In this paper, the feature input vector is a binary sparse vector by one-hot encoding the identity of a user or an item. Above the input is the embedding layer, which transforms the sparse feature vector into a dense embedding vector. After that, a user or an item will be represented by a dense vector, also known as $embedding\ vector$. Next, the user and item embedding vectors will be fed into the interaction function to map

the latent vectors to prediction scores. Then let $\widehat{r_{ui}}$ denote the predicted preference score of user u on item i. The calculation of $\widehat{r_{ui}}$ in recommender system is formulated as follows:

$$\widehat{r_{ui}} = f(\mathbf{p}_u, \mathbf{q}_i \mid \mathbf{P}, \mathbf{Q}, \Theta_f) \tag{1}$$

where $f(\cdot)$ is the interaction function modelling the user u's preference on item i and Θ_f is the parameters of it. We define that $\mathbf{P} = \{\mathbf{p}_u\}_{u \in U}$ denotes the embedding matrix of users, $\mathbf{Q} = \{\mathbf{q}_i\}_{i \in I}$ denotes the embedding matrix for items. Let $\mathbf{p}_u = \mathbf{P}^T v_u^U$ ($\mathbf{p}_u \in \mathbb{R}^D$) and $\mathbf{q}_i = \mathbf{Q}^T v_i^I$ ($\mathbf{q}_i \in \mathbb{R}^D$), denoting the embedding vector for user u and item i, respectively, and D is the dimension of embedding vectors.

NCF [15] is a deep CF framework proposed to capture the complex and nonlinear relationships between users and items with multilayer perceptrons (MLP). GMF is one of the most representative collaborative filtering model constructed under this framework. In GMF, pairwise user embedding and item embedding will be mapped to preference scores by utilizing both MLP and the inner product.

$$\begin{aligned} \phi_1(\mathbf{p}_u, \mathbf{q}_i) &= \mathbf{p}_u \odot \mathbf{q}_i \\ \widehat{r_{ui}} &= \phi_{out}(\phi_x ... (\phi_2(\phi_1(\mathbf{p}_u, \mathbf{q}_i)))) \end{aligned} \tag{2}$$

where ϕ_1 is the the inner product between the user and the item embedding vector, ϕ_x and ϕ_{out} are the x-th and output MLP layer of GMF.

3.2 Bayesian Personalized Ranking

In this paper, we use the Bayesian Personalized Ranking (BPR) loss function, which is a widely used pairwise loss function for optimizing recommender systems towards personalized ranking [24]. The basic intuition is that observed (positive) items should be ranked higher than the unobserved (negative) ones. To implement this idea, the BPR objective function is formulated as follow:

$$L_{BPR} = \sum_{(u,i,j) \in T} -ln\sigma(\widehat{r_{ui}} - \widehat{r_{uj}}) + \lambda \left\| \Theta \right\|^2 \tag{3}$$

where Θ denotes the parameters of the model including the embedding matrix $\widehat{\mathbf{\Theta}}$ and Θ_f, λ controls the importance of regularization parameters to prevent overfitting and $\sigma(\cdot)$ is the sigmoid function. The set $T = \left\{(u,i,j) \mid u \in U, i \in I_u^+, j \in I \setminus I_u^+\right\}$ denotes the set of all pairwise training instances, where I_u^+ denotes the set of observed items of user u. As we can see, by optimizing the BPR loss, we obtain a larger margin between observed items and the unobserved ones and then get the personalized ranking list for user u.

4 Method

In this section, we first introduce the architecture of Adversarial training GMF (AGMF). Next, we present the DAGMF method, an instantiation of DAT combined with GMF. Lastly, the strategy of choosing the direction for perturbation is discussed in detail.

4.1 Adversarial Training

Adversarial training is proposed by [12] as a novel regularization method for improving the robustness of the classifier model in CV domain. Unlike in CV domain, [22] applies adversarial training in NLP domain no longer adding the perturbation in input space. They add the adversarial perturbation in the embedding layer and consider this method as effective regularization to prevent overfitting and improve generalization performance. Similar to the language processing, APR [14] introduce the adversarial training in MF-BPR, via perturbing the embedding vectors which represent the users and items. In this paper, we would like to extend APR in the deep learning model, GMF, with exerting the perturbation only in the embedding layer, namely AGMF.

In this section, we first introduce the architecture of Adversarial training GMF (AGMF). Next, we present the DAGMF method, an instantiation of DAT combined with GMF. Lastly, the strategy of choosing the direction for perturbation is discussed in detail.

Let $\mathbf{\Delta}_{adv}^u$ denote the adversarial perturbation to be added to the embedding vector of user u. The dimension size of $\mathbf{\Delta}_{adv}^u$ is D, which is the same as the user u embedding vector. Let $\mathbf{p_\Delta}_{adv}^u$ denote the adversarial embedding vector of user u ($\mathbf{p_\Delta}_{adv}^u = \mathbf{p}_u + \mathbf{\Delta}_{adv}^u$). Analogously, we can obtain the adversarial embedding vector $\mathbf{q_\Delta}_{adv}^i$ for item i. The predicted preference scores of user u in item i can be calculated:

$$\widehat{r_{ui\Delta}} = f(\mathbf{p_\Delta}_{adv}^u, \mathbf{q_\Delta}_{adv}^i \mid \mathbf{P}_{\mathbf{\Delta}adv}, \mathbf{Q}_{\mathbf{\Delta}adv}, \Theta_f) \tag{4}$$

where $\mathbf{P}_{\mathbf{\Delta}adv} = \{\mathbf{p_\Delta}_{adv}^u\}_{u \in U}$ denotes the adversarial embedding matrix for users, $\mathbf{Q}_{\mathbf{\Delta}adv} = \left\{\mathbf{q_\Delta}_{adv}^i\right\}_{i \in I}$ denotes the adversarial embedding matrix for items.

The adversarial perturbation aims to maximize the objective function of recommendation model. Thus we define it as follow:

$$\mathbf{\Delta}_{adv} = \underset{\Delta, \|\Delta\| \leq \epsilon}{\arg\max} L_{BPR}(T \mid \widehat{\mathbf{\Theta}} + \mathbf{\Delta}, \Theta_f) \tag{5}$$

where $\widehat{\mathbf{\Theta}}$ is the embedding matrix for users and items, including $\mathbf{P}$ and $\mathbf{Q}$, $\mathbf{\Delta}$ denotes the perturbation on embedding matrix, and ϵ represents the hyper-parameter that controls the norm of $\mathbf{\Delta}$. T denotes the set of all pairwise training instances. In this paper, we adapt L_2 norm ($\|\cdot\|$).

It is intractable to get the estimate $\mathbf{\Delta}_{adv}$ in the Eq.6 because the objective function and the neural network involve the sophisticated operations. As such, [12] proposes an approximation method fast gradient sign by linearizing the objective function around $\mathbf{\Delta}$. With this method and the ϵ-constraints, the $\mathbf{\Delta}_{adv}$ can be further approximated as follow:

$$\mathbf{\Delta}_{adv} = \epsilon \frac{\mathbf{g}}{\|\mathbf{g}\|}, \mathbf{g} = \nabla_{\widehat{\mathbf{\Theta}}} L_{BPR}(T \mid \widehat{\mathbf{\Theta}} + \mathbf{\Delta}, \Theta_f) \tag{6}$$

After obtaining the $\mathbf{\Delta}_{adv}$, the adversarial objective function can be defined as:

$$L_{BPR\Delta} = \sum_{(u,i,j) \in T} -ln\sigma(\widehat{r_{ui\Delta}} - \widehat{r_{uj\Delta}}) \tag{7}$$

In the end, the adversarial training objective function is the combination of L_{BPR} and $L_{BPR\Delta}$:

$$L_{adv} = L_{BPR} + \lambda L_{BPR\Delta} \tag{8}$$

This formulation will be minimized in the training process. The adversarial term $L_{BPR\Delta}$ can be seen as an adversarial regularizer. λ controls the relative importance of it, and when the $\lambda = 0$, adversarial term has no impact on training.

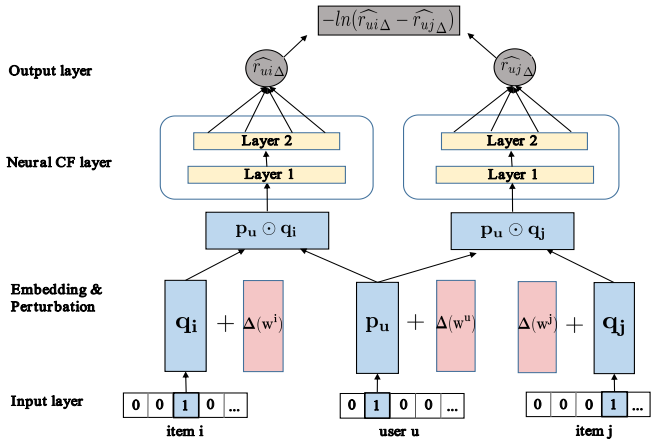


Figure 3. Illustration of our DAGMF method. The perturbations $\mathbf{\Delta(w)}$ are enforced on each embedding vector of user and item.

4.2 Directional Adversarial Training

However, only considering perturb toward the maximum direction in the embedding space for recommendation models will bring some noise information into the training process. As such, we propose Directional Adversarial Training to address this problem for better applying adversarial training in recommender systems. The intuition of our method is that the perturbation direction in embedding space can be restricted toward other examples in existing embedding space instead of the worst perturbation direction so that we can integrate the collaborative signal into the training process. In playing the minimax game, the quality of embedding layers will be improved gradually. Figure 3 illustrates the framework of applying Directional Adversarial Training in the model GMF.

Firstly, we define the direction vector in the embedding space from user u_t to user u_z as $\mathbf{d}_{u_z}^{u_t}$:

$$\mathbf{d}_{u_z}^{u_t} = \frac{\widetilde{\mathbf{d}}_{u_z}^{u_t}}{\left\|\widetilde{\mathbf{d}}_{u_z}^{u_t}\right\|_2}, \widetilde{\mathbf{d}}_{u_z}^{u_t} = \mathbf{p}_{u_z} - \mathbf{p}_{u_t}, \tag{9}$$

It is worth noting that the direction vector $\mathbf{d}_{u_z}^{u_t}$ is always a unit vector, $\|\mathbf{d}_{u_z}^{u_t}\|_2 = 1$. Specially, if the $z = t$, the $\mathbf{d}_{u_z}^{u_t}$ will be defined as a zero vector.

Next, we define $w_{u_z}^{u_t}$ as the weight corresponding to the direction vector $\mathbf{d}_{u_z}^{u_t}$. Next let $\mathbf{\Delta}(w^u)$ denote the the direction adversarial perturbation of user u:

$$\mathbf{\Delta}(w^u) = w_{u'}^u \mathbf{d}_{\mathbf{u'}} \tag{10}$$

In our method, the users which training example will perturb towards is defined as the target set U_{target}^u. Thus, u' is from the set of U_{target}^u.

Then, the directional adversarial embedding vector of user u is formulated as follow:

$$\mathbf{p}_{\Delta_{dadv}^u} = \mathbf{p}_u + \mathbf{\Delta}(w^u) \tag{11}$$

Since BPR loss will be minimized in the training process, we consider how to seek the worst-case weights of the direction vectors that can maximize the BPR loss:

$$w_{dadv}^u = \underset{w^u, \|w^u\| \leq \epsilon}{\arg\max} L_{BPR}(T \mid \widehat{\mathbf{\Theta}} + \mathbf{\Delta}(w^u), \Theta_f) \tag{12}$$

Algorithm 1 SGD learning algorithm for DAGMF

Input: Training data T
Parameter: embedding size, learning rate η, ladversarial noise level ϵ, adversarial strength λ
Output: $\widehat{\mathbf{\Theta}}, \Theta_f$

1: Initialize $\widehat{\mathbf{\Theta}}, \Theta_f$ from GMF
2: **while** criteria not converge **do**
3: Randomly sample examples (u, i, j) from D
4: // Calculating the direction vector
5: $\mathbf{d} \leftarrow$ Equation (9)
6: // Calculating the worst case weights
7: $w_{dadv} \leftarrow$ Equation (12)
8: // Constructing direction adversarial perturbation
9: $\mathbf{\Delta}(w) \leftarrow$ Equation (10)
10: //Optimizing model parameters
11: $\widehat{\mathbf{\Theta}}, \Theta_f \leftarrow$ Equation(14)
12: **end while**

The approximation method in the Eq.6 is no longer performed in calculating the worst-case weights after we restrict the direction of perturbation. Therefore, we borrow the idea from the [23], in which the perturbation strength w^u can be approximated estimated by applying the second-order Taylor in $L_{BPR}(T \mid \widehat{\mathbf{\Theta}} + \mathbf{\Delta}(w^u), \Theta_f)$. Formally, the solution to estimate worst-case weight can be defined as follow:

$$w_{dadv}^u = \epsilon \frac{g}{\|g\|}, g = \nabla_{w^u} L_{BPR}(T \mid \widehat{\mathbf{\Theta}} + \mathbf{\Delta}(w^u), \Theta_f) \tag{13}$$

Analogously, w_{dadv}^i can be obtained by following the process mentioned above. Similar to Eq.8, the overall objective function for directional adversarial training is optimized as follow:

$$< \widehat{\mathbf{\Theta}}, \Theta_f >= \underset{\widehat{\mathbf{\Theta}}, \Theta_f}{\arg\min} L_{BPR} + \lambda L_{BPR_{\Delta_{dadv}}} \tag{14}$$

By unifying the two process above, we can formulate a minimax objective function. The optimization of model parameters $< \widehat{\mathbf{\Theta}}, \Theta_f >$ is the minimizing player, and seeking the worst-case perturbations weights w_{dadv} is the maximizing player:

$$< \widehat{\mathbf{\Theta}}, \Theta_f, w_{dadv} >= \underset{\widehat{\mathbf{\Theta}}, \Theta_f}{\arg\min} \underset{w_{dadv}}{\arg\max} L_{BPR} + \lambda L_{BPR_{\Delta_{dadv}}} \tag{15}$$

From another point of view, our DAT can be regarded as a way of data augmentation, which trains the model on both raw data and perturbation data with containing collaborative signal simultaneously. We leave this exploration as future work since we focus on adversarial training in this paper.

Algorithm 1 shows the detailed training procedure of our DAGMF. Finally, it is worth noting that in the beginning, the model parameters $\widehat{\mathbf{\Theta}}, \Theta_f$ are initialized by optimizing GMF (line 1), instead of randomly. This is because when the model is underfitting, the normal training process is sufficient to get better parameters. It is necessary to add the adversarial perturbations after the model parameters start to overfit the data.

4.3 The Strategy of Choosing the Direction

As mention above, the worst direction of perturbation is abandoned, and the directions of perturbations are restricted inside the existing

embedding space with learnable weight so that we avoid the incorrect information incorporated in the training process. In addition, with choosing the perturbation direction properly, the collaborative signal can be encoded in the directions of perturbations in embedding space, which subsequently enhance the expressive of embedding layers. As we all know, the core assumption of Collaborative Filtering recommender systems is that the users with similar behaviours and features have similar preference.

Following this assumption, we instinctively restrict the direction of perturbation for user u to the Top-K nearest neighbours of user u in the users embedding space. We fix the Top-K nearest neighbours of user u, $n^{(K)}$, as U^u_{target}, based on the pre-trained model parameters for improving model efficiency. Firstly, we randomly select a target user u' ($u' \in U^u_{target}$) to calculate the direction vector from user u to u'. Then, the weight $w^u_{u'}$ can be obtained according to the Eq.12. Moreover, the direction of perturbation for observed item i will be selected from the $I^+_u - i$. Likewise, the target set I^j_{target} of unobserved item j will be obtained by substitute $I^+_u - i$ with $I \setminus I^+_u$. The weight $w^i_{i'}$ or $w^j_{j'}$ will be obtained in the same way of the user. To summarize, the advantage of our DAT is that it can incorporate the collaborative signal into CF recommender systems. Through this way, the preferences between similar users would be captured precisely in the training process, which benefits the performance of recommender systems significantly.

Table 1. Dataset

Datasets	#users	#items	#ratings	sparsity
Yelp	2,265	11,386	44,812	99.82%
Movielens	6,040	3,952	1,000,000	95.72%
Ciao	7,375	105,114	284,086	99.97%

5 Experiments

In this section, we perform experiments on three real-world datasets to evaluate our proposed method with the aim of answering the following research questions:

RQ 1 Can our proposed DAGMF outperform the state-of-the-art traditional and adversarial recommendation methods?

RQ 2 How is the effect of the adversarial training and can it improve the generalization of the model?

RQ 3 How do the key hyper-parameters ϵ and λ affect the performance? how to choose the optimal values?

In what follows, we first describe the experimental settings, followed by answering the above three research questions in turn.

5.1 Experimental Settings

Dataset. To evaluate the effectiveness of DAGMF, we conduct experiments on three public datasets: Yelp[4], MovieLens[5], and Ciao[27], which are accessible and vary in terms of domain, size, and sparsity. Especially, as the Yelp dataset is too large, we randomly select 3,000 users and discard the users with less than two interactions and the items without interactions. Table 1 summarizes their detailed statistics. For each dataset, we treat items that have interaction (e.g., purchase, click) with the user as observed items and the others as unobserved items.

[4] https://github.com/hexiangnan/theano-BPR
[5] https://grouplens.org/datasets/movielens/

Evaluation Protocols. Following the prominent work in item ranking recommendation [16, 2, 24], we adopt the standard leave-one-out protocol to evaluate the performance of models. Specifically, we hold-out the latest interaction as the test set for Movielens and Yelp datasets and randomly select one interaction item as the test set for Ciao dataset because they don't have timestamp. The remaining data will be maintained for training. While it is too time-consuming to ranking all items for every user during the evaluation, we follow the common approach [10, 18] to randomly sample 499 items that have not interacted with the user. Then, we rank the test item with these 499 items as the rank list. We study the performance of Top-N recommendation models with HR and NDCG. Without special mention, we truncate the ranking list of 500 items at position 10 for both metrics. Therefore, the HR@10 intuitively judges whether the Top-10 list contains the test items and the NDCG@10 measures position of the test item in the Top-10 list, assigning the higher scores for the higher position.

Baselines. We compare our DAGMF with the following methods.

- **ItemPop.** This is a non-personalized method, which ranks the items according to their popularity evidenced by the number of interactions in the training set. It benchmarks the performance of the personalized recommendation.
- **MFBPR [24].** In this method, the MF model is optimized by the BPR loss in the Eq.3. It is a baseline method for personalized ranking recommendation and only exploits the linear interaction between users and items.
- **GMF [15].** GMF is a state-of-the-art neural CF model which combines the linearity of inner product and non-linearity of multiple hidden layers for modelling user-item interactions. Specially, we employ the one-layer MLP.
- **AGMF [14].** This method is constructed by encoding the GMF into the adversarial training for the recommendation (APR) [14]. APR refers to the whole mechanism in image processing. It works as a regularization by perturbing the embedding layer of examples towards the maximize direction.
- **NGCF [30].** This method learns the more effective embedding layer by encoding the collaborative signal in the form of high-order connectivities which performs via embedding propagation. We use the two-order propagation to implement the NGCF and follow the learning rate and the L_2 normalization in this paper.
- **IRGAN [28].** In IRGAN, a generative model approximates the relevance distribution to generate user-item pairs and feed them together with the pairs constructed from the real data to the discriminator and then the discriminative model tries to classify the real data and the data generated from the generative model. We use the code released by the authors. For model initialization, we employ the pre-train embedding layer in LambdaFM [32] for the generator following the suggestion in [14].

The aforementioned baselines are the state-of-the-art model-based CF for item ranking recommendation. GMF is the advanced deep neural collaborative filtering model which has achieved a significant improvement above the conventional model (e.g., MFBPR, item-based CF). AGMF and IRGAN make use of the adversarial training to improve the training process so they show outstanding performance. NGCF exploits the high-order collaborative signal through the embedding propagation. CFGAN [4] is the vector-wise GAN-based CF without learning the embedding vectors so we do not select it for comparison.

Implementation details. DAGMF model is implemented based on PyTorch. For a fair comparison, we fix the embedding

Table 2. Top-K recommendation performance comparison of different methods. The Tra. and Adv. indicates that the relative improvement of DAGMF over the best traditional methods and adversarial methods.

Datasets	Metrics	Traditional				Adversarial		Ours		Improvement	
		ItemPop	MFBPR	GMF	NGCF	IRGAN	AGMF	DAGMF-R	DAGMF	Tra.	Adv.
MovieLens	HR@5	0.1222	0.2214	0.2404	0.2591	0.2424	0.2495	0.2437	**0.2638**	1.814%	5.731%
	HR@10	0.1892	0.3204	0.3573	0.3653	0.3584	0.3608	0.3561	**0.3729**	2.080%	3.354%
	NDCG@5	0.0740	0.1354	0.1462	0.1501	0.1454	0.1533	0.1480	**0.1622**	8.061%	5.806%
	NDCG@10	0.0946	0.1658	0.1818	0.1894	0.1806	0.1840	0.1823	**0.1913**	1.003%	3.967%
Ciao	HR@5	0.0275	0.2369	0.2511	0.2758	0.2568	0.2715	0.2487	**0.2868**	3.988%	5.635%
	HR@10	0.1223	0.2951	0.3137	0.3285	0.3079	0.3245	0.3033	**0.3439**	4.688%	5.978%
	NDCG@5	0.0105	0.1664	0.1813	0.1994	0.1902	0.1979	0.1802	**0.2093**	4.965%	5.760%
	NDCG@10	0.0510	0.1842	0.2005	0.2164	0.2058	0.2141	0.1980	**0.2298**	6.192%	7.333%
Yelp	HR@5	0.1506	0.7377	0.7638	0.8021	0.7909	0.7978	0.7475	**0.8199**	3.512%	2.767%
	HR@10	0.2283	0.8406	0.8637	0.8957	0.8720	0.8862	0.8667	**0.8938**	0.915%	-0.278%
	NDCG@5	0.0948	0.5352	0.5728	0.5953	0.5916	0.6022	0.5471	**0.6355**	6.742%	5.519%
	NDCG@10	0.1185	0.5610	0.6041	0.6344	0.6257	0.6328	0.5891	**0.6617**	4.293%	4.568%

size to 32 for all the models and optimize them (except IRGAN) via the mini-batch Adagrad [9] with the batch size in $\{256, 512, 1024\}$. For IRGAN, we set the optimizer to stochastic gradient descent (SGD) following the code released by the authors. For hyperparameters, we apply the grid search: the learning rate η is tuned in $\{0.001, 0.005, 0.01, 0.05\}$, the coefficient of L_2 normalization is tuned in $\{0.0001, 0.001, 0.01, 0.1\}$. For DAGMF, the λ is tune in $\{0.001, 0.01, 0.1, 1, 10, 100\}$, ϵ is tune in $\{0.1, 0.3, 0.5, 0.7, 0.9, 1, 5\}$ and the K in $n_{(K)}$ is fixed at 20.

5.2 Performance Comparison

In this section, we compare the HR@N and NDCG@N of DAGMF with all baselines, where N $\in \{5, 10\}$. The results are shown in Table 2. Inspecting the result from the top to bottom, we can obtain the following key observations:

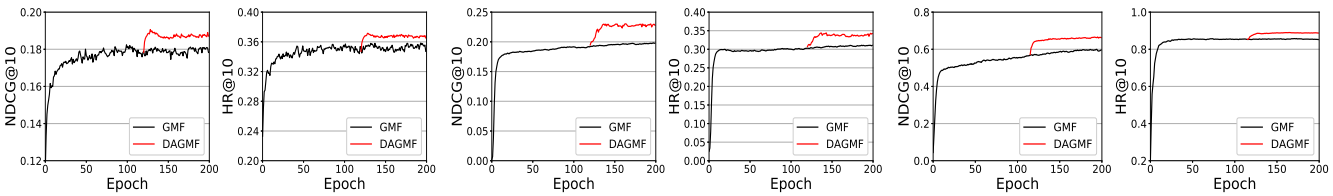
- Our DAGMF outperforms the-state-of-the-art adversarial CF model (e.g., IRGAN and AGMF) on three datasets and traditional CF model (e.g., IRGAN and AGMF) in most case with a p-value of smaller than 0.01. The only exception is on Yelp dataset, in which the NGCF obtains better performance than DAGMF by a small margin in HR@10.
- DAGMF-R is the method, in which we randomly pick the examples without involving the collaborative signal. Specifically, the U^u_{target} is set to all other users instead of the Top-K nearest neighbours. The target items of observed items and the unobserved items are chosen from all other items rather than only from the items with the simple label. We can observe that the performance of DAGMF-R is much worse than DAGMF, which verifies the importance of the collaborative signal. Moreover, it is worth noting that DAGMF-R even achieves poorer performance than GMF in sparse datasets (Ciao and yelp). We analyze that in sparse datasets, more incorrect information is encoded in the DAGMF-R model when the direction of perturbation is chosen randomly. This crashes the performance of DAGMF-R model.
- Compared with the AGMF, a recently proposed adversarial training model, DAGMF avoids encoding the error information into training and exhibits an average improvement of 4.1%. Such significant improvement might be attributed to that DAGMF incorporates the interaction between similar users or items in adversarial training.
- DAGMF generally performs better than IRGAN, which applies a generator model and a discriminative model to play a minimax game. This again verifies the effective of our DAT. Another advantage of DAT is that unlike the GAN model need to tune the complicate structure and hyper-parameters to overcome the model-collapse, DAT has succinct architecture and only requires the initialization parameters of GMF.
- NGCF is the-state-of-the-art model-based CF, which exploits the collaborative signal into the embedding layer. We find that DAGMF is consistently superior to NGCF, and the improvement of it is 3.4%. This might be that applying the high-order embedding propagation will lead to the overfitting problem. This phenomenon has already been pointed out by the authors in [30].

5.3 Impact of Directional Adversarial Training Regularization

In this section, we explore the impact of DAT on the generalization performance of GMF, from two aspects: the training epoch and the embedding size.

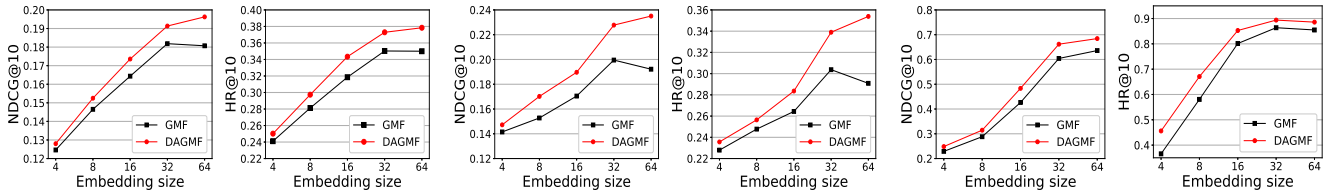
Training Process. We show the training process of GMF and DAGMF: Firstly GMF will be trained for 120 epochs and then we perform the DAT on parameters from the GMF for 80 epochs. Note here that we evaluate their generalization performance per epoch on three datasets and the results are shown in the Figure 4. As can be observed, GMF converges after 120 training epochs, whereas further training GMF with DAT can bring an obvious improvement in HR and NDCG. To be specific, on the Ciao dataset, the best HR and NDCG of GMF is 0.3137 and 0.2005, respectively, which are further improved to 0.3439 and 0.2298 by DAGMF. There is roughly 9.6% and 14.6% relative improvement in HR and NDCG. These results again verify the effectiveness of DAGMF, emphasizing the significance of DAT to improve model performance.

Embedding Size. Furthermore, we study how the embedding size influences the model performance and the result is shown in the Figure 5. Under the different embedding size, our DAGMF consistently outperforms the GMF. Note here that when the embedding size is 4, the performance improvement of DAGMF becomes less significant. The probably reason is that the small capacity of the model constrains the information that the embedding layer can contain. Such that the advantage of DAT can not be exploited. In addition, we can notice that as the embedding size becomes larger, the GMF becomes slightly overfitting, especially in the Ciao dataset. Specially, the performance of GMF degrades, while the DAGMF increase continually when the embedding size is fixed at 64. This illustrates that DAT is



(a) MovieLens (b) Ciao (c) Yelp

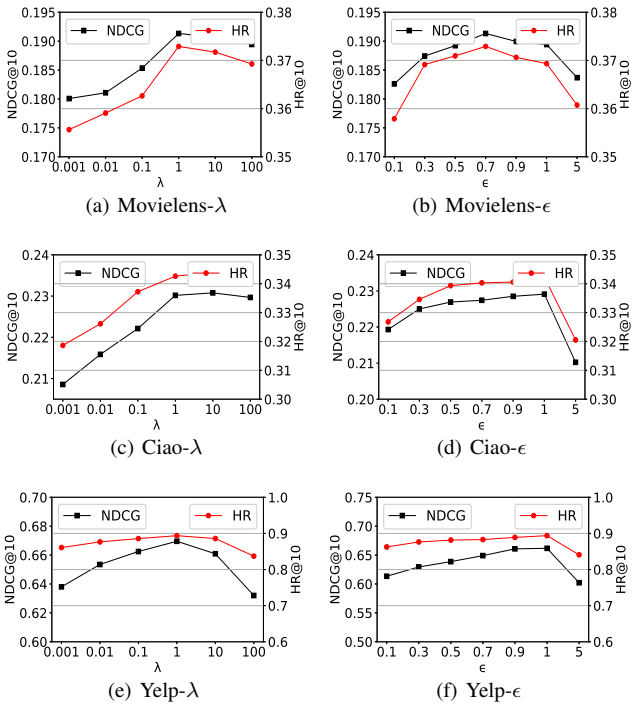
Figure 4. Learning curves of GMF and DAGMF.



(a) MovieLens (b) Ciao (c) Yelp

Figure 5. Impact of embedding size on GMF and DAGMF.

an effective regularization to prevent the overfitting for model-based CF. In addition, it is worth noting that HR and NDCG of DAGMF exhibit different trends when the embedding size is set to 64 in the Yelp dataset. This is interpretable since BPR loss takes advantage in ranking top items and it can rank the test items in higher position after they are in the Top-10 list.



(a) Movielens-λ (b) Movielens-ϵ

(c) Ciao-λ (d) Ciao-ϵ

(e) Yelp-λ (f) Yelp-ϵ

Figure 6. Impact of λ and ϵ.

5.4 Hyper-parameter Sensitivity

Because of the succinct architecture of DAGMF, it only introduces two additional hyperparameters λ and ϵ to control the strength of directional adversarial regularization and the perturbation level, respectively. Figure 6 shows the HR and NDCG of DAGMF varying the different parameters on three datasets. To explore the optimal value of one parameter, the other parameters will be fixed in the same.

Figure 6 (a), (c) and (e) displays the performance trend of DAGMF by varying the λ from 0.001 to 100. As can be seen, when the λ is smaller than the threshold (i.e,1 in Movielens dataset), the performance of DAGMF will increase as the λ becomes larger. When the λ is larger than the threshold, which will lead to performance degradation. The threshold value is different across the different datasets, 1 for Yelp and MovieLens, 10 for Ciao.

Figure 6 (b), (d) and (f) shows the performance trend with respect to ϵ, which is from the set $\{0.1, 0.3, 0.5, 0.7, 0.9, 1, 5\}$. The optimal results are obtained when $\epsilon = 0.7$ on MovieLens, $\epsilon = 1$ on Ciao and Yelp. We can observe that when ϵ is set too small, the DAGMF only has minor improvements. The possible reasons lies in that the DAGMF is almost the same as GMF. In addition, when the ϵ is set too large, the perturbation will ruin the training process of model and cause the performance drop.

6 Conclusion

In this work, we propose a novel technique, Directional Adversarial Training, by restricting the perturbation direction towards existing examples in the embedding space to address the limitation of existing adversarial training model-based CF that will change the original semantic information hidden in the user-item interaction. Moreover, we newly design the strategy to incorporate collaborative signal into the perturbation direction, which explicitly learns the more effective embedding for the model. We apply the directional adversarial training as regularization in a recently developed neural model-based CF, GMF. Extensive experiments on three real-world datasets are conducted to prove the rationality and effectiveness of DAGMF.

In the future, we would like to explore how to employ the directional adversarial training technique on the deeply hidden layer, whose parameters also can be added with perturbation. In addition, we are interested in extending the adversarial training technique to other recommendation models, such as content-based [5], session-based [17]. Last but not the least, we think the adversarial training technique has the potential for other information retrieval scenarios such as text retrieval [1], web search [3] and question answering [25].

ACKNOWLEDGEMENTS

This work supported by the National Natural Science Foundation of China (61702568,U1711267), the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (2017ZT07X355) and the Fundamental Research Funds for the Central Universities under Grant (17lgpy117).

REFERENCES

[1] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al., *Modern information retrieval*, volume 463, ACM press New York, 1999.

[2] Immanuel Bayer, Xiangnan He, Bhargav Kanagal, and Steffen Rendle, 'A generic coordinate descent framework for learning from implicit feedback', in *Proceedings of the 26th International Conference on World Wide Web*, pp. 1341–1350, (2017).

[3] Christopher Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N Hullender, 'Learning to rank using gradient descent', in *Proceedings of the 22nd International Conference on Machine learning (ICML-05)*, pp. 89–96, (2005).

[4] Dong-Kyu Chae, Jin-Soo Kang, Sang-Wook Kim, and Jung-Tae Lee, 'Cfgan: A generic collaborative filtering framework based on generative adversarial networks', in *Proceedings of the 27th ACM international conference on information and knowledge management*, pp. 137–146. ACM, (2018).

[5] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua, 'Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention', in *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 335–344. ACM, (2017).

[6] Liang Chen, Yang Liu, Xiangnan He, Lianli Gao, and Zibin Zheng, 'Matching user with item set: collaborative bundle recommendation with deep attention network', in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 2095–2101. AAAI Press, (2019).

[7] Liang Chen, Angyu Zheng, Yinglan Feng, Fenfang Xie, and Zibin Zheng, 'Software service recommendation base on collaborative filtering neural network model', in *Service-Oriented Computing - 16th International Conference, ICSOC 2018, Hangzhou, China, November 12-15, 2018, Proceedings*, pp. 388–403, (2018).

[8] Quanyu Dai, Xiao Shen, Liang Zhang, Qiang Li, and Dan Wang, 'Adversarial training methods for network embedding', in *The World Wide Web Conference*, pp. 329–339. ACM, (2019).

[9] John Duchi, Elad Hazan, and Yoram Singer, 'Adaptive subgradient methods for online learning and stochastic optimization', *Journal of Machine Learning Research*, **12**(Jul), 2121–2159, (2011).

[10] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He, 'A multi-view deep learning approach for cross domain user modeling in recommendation systems', in *Proceedings of the 24th International Conference on World Wide Web*, pp. 278–288, (2015).

[11] Fuli Feng, Xiangnan He, Jie Tang, and Tat-Seng Chua, 'Graph adversarial training: Dynamically regularizing based on graph structure', *arXiv preprint arXiv:1902.08226*, (2019).

[12] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy, 'Explaining and harnessing adversarial examples', *arXiv preprint arXiv:1412.6572*, (2014).

[13] Lei Guo, Hongzhi Yin, Qinyong Wang, Tong Chen, Alexander Zhou, and Nguyen Quoc Viet Hung, 'Streaming session-based recommendation', in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1569–1577. ACM, (2019).

[14] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua, 'Adversarial personalized ranking for recommendation', in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 355–364. ACM, (2018).

[15] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua, 'Neural collaborative filtering', in *Proceedings of the 26th international conference on world wide web*, pp. 173–182, (2017).

[16] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua, 'Fast matrix factorization for online recommendation with implicit feedback', in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 549–558. ACM, (2016).

[17] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk, 'Session-based recommendations with recurrent neural networks', *arXiv preprint arXiv:1511.06939*, (2015).

[18] Yehuda Koren, 'Factorization meets the neighborhood: a multifaceted collaborative filtering model', in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 426–434. ACM, (2008).

[19] Yehuda Koren, Robert Bell, and Chris Volinsky, 'Matrix factorization techniques for recommender systems', *Computer*, (8), 30–37, (2009).

[20] U Liji, Yahui Chai, and Jianrui Chen, 'Improved personalized recommendation based on user attributes clustering and score matrix filling', *Computer Standards & Interfaces*, **57**, 59–67, (2018).

[21] Greg Linden, Brent Smith, and Jeremy York, 'Amazon. com recommendations: Item-to-item collaborative filtering', *IEEE Internet computing*, (1), 76–80, (2003).

[22] Takeru Miyato, Andrew M Dai, and Ian Goodfellow, 'Adversarial training methods for semi-supervised text classification', *arXiv preprint arXiv:1605.07725*, (2016).

[23] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii, 'Virtual adversarial training: a regularization method for supervised and semi-supervised learning', *IEEE transactions on pattern analysis and machine intelligence*, **41**(8), 1979–1993, (2018).

[24] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme, 'Bpr: Bayesian personalized ranking from implicit feedback', in *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pp. 452–461. AUAI Press, (2009).

[25] Cicero dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou, 'Attentive pooling networks', *arXiv preprint arXiv:1602.03609*, (2016).

[26] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus, 'Intriguing properties of neural networks', *arXiv preprint arXiv:1312.6199*, (2013).

[27] Jiliang Tang, Huiji Gao, and Huan Liu, 'mtrust: discerning multi-faceted trust in a connected world', in *Proceedings of the fifth ACM international conference on Web search and data mining*, pp. 93–102. ACM, (2012).

[28] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang, 'Irgan: A minimax game for unifying generative and discriminative information retrieval models', in *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 515–524. ACM, (2017).

[29] Xiang Wang, Xiangnan He, Liqiang Nie, and Tat-Seng Chua, 'Item silk road: Recommending items from information domains to social users', in *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 185–194. ACM, (2017).

[30] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua, 'Neural graph collaborative filtering', *arXiv preprint arXiv:1905.08108*, (2019).

[31] Fenfang Xie, Shenghui Li, Liang Chen, Yangjun Xu, and Zibin Zheng, 'Generative adversarial network based service recommendation in heterogeneous information networks', in *2019 IEEE International Conference on Web Services (ICWS)*, pp. 265–272. IEEE, (2019).

[32] Fajie Yuan, Guibing Guo, Joemon M Jose, Long Chen, Haitao Yu, and Weinan Zhang, 'Lambdafm: learning optimal ranking with factorization machines using lambda surrogates', in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pp. 227–236. ACM, (2016).