BiGAN: Collaborative Filtering with Bidirectional Generative Adversarial Networks

Chapter · January 2020

DOI: 10.1137/1.9781611976236.10

Ding Rui
Northeastern University (Shenyang, China)

Bowei Chen
Carnegie Mellon University

BiGAN: Collaborative Filtering with Bidirectional Generative Adversarial Networks

Rui Ding* Guibing Guo⊠† Xiaochun Yang⊠‡ Bowei Chen§ Zhirong Liu ¶ Xiuqiang He‖

Abstract

Recently, GAN-based collaborative filtering methods have gained increasing attention in recommendation tasks which can learn remarkable user and item representation. However, these existing GAN-based methods mainly suffer from two limitations: (1) Their trainings are not comprehensive given the fact that the discriminator may be trained misleadingly and over-early converging since the generator may accidentally sample real items as fake ones, resulting in the emergence of contradicting labels for the same items. (2) They fail to consider implicit friends (users with the same interests.), leading to severe limitations of recommendation performance. In this paper, we propose BiGAN, an innovative bidirectional adversarial recommendation model which can alleviate the limitations mentioned above in recommendation tasks. It consists of two GANs, namely ForwardGAN and BackwardGAN. Specifically, ForwardGAN learns to generate a group of possible interacted items given a specific user, it aims to ensure that the discriminator D_f can be trained effectively. Furthermore, BackwardGAN fully exploits implicit friends with similar behaviors, then propagates them back to ForwardGAN, where a similarity exploration strategy is implemented to gain more outstanding user representation. Therefore, two GANs are trained jointly in a circle, where the augment of one GAN will enhance another one, leading to the promising user and item representation. In the experimental part, we demonstrate that our model is superior to other state-of-the-art recommenders.

1 Introduction

Recommender systems play a pivotal role in providing friendly and satisfying user experience in many scenarios such as online media and retail platforms, seeking to alleviate the issue of information overload for active users [1, 3]. Recently, among different recommendation strategies, popular collaborative filtering (CF) methods have achieved significant success which personalize item recommendations based on historic interactions between users and items [4].

Since Generative Adversarial Networks (GAN) [5] was proposed, CF-based methods utilizing GAN have been proven to be capable of exploring high-quality user and item representation [3, 6, 7, 8, 9] to boost the ranking performance. However, they all suffer from some limitations, especially data sparsity and cold start (only regard historic interactions as independent input). For example, IRGAN [9], a famous GAN-based model for recommendation, attempts to sample a single discrete item index to train the network. However, the training scheme makes generator sample data which may be exactly the same as those in the ground truth, leading to discriminator over-early converging with a large percentage of contradicting labels, i.e., one same item is sometimes marked as "fake" (generated) while sometimes as "real" (ground truth). The subfigure (a) in Figure 1 illustrates this issue. This contradicting problem will become more severe when the item pool is super small since the free-of-contradicting items are less in the candidate sampling pool. CFGAN [8] tries to ameliorate the issue mentioned above by introducing a real-valued vector-wise adversarial training. However, it requires really high space complexity, which is inefficient and costly in a real-world application. What's more, the more items, the more difficult the training and the less efficient. In conclusion, existing methods either suffer from the issue of unpromising model training caused by contradicting labels, or require a large space to gain exhaustive exploration of user and item representation. More importantly, they all fail to analyze the implicit similarity between diversified users with same behaviors [11], which will in turn enhance user representation by exploiting implicit friendships and alleviate the limitations of sparsity and cold start in recommendation

*School of Computer Science and Engineering, Northeastern University, China. ruiding.neu@outlook.com.

†School of Software, Northeastern University, China. guogb-@swc.neu.edu.cn.

‡School of Computer Science and Engineering, Northeastern University, China. yangxc@mail.neu.edu.cn

§School of Software, Northeastern University, China. boweichen_public@outlook.com

¶Huawei Noah's Ark Lab, China. liuzhirong@huawei.com

‖Huawei Noah's Ark Lab, China. hexiuqiang1@huawei.com

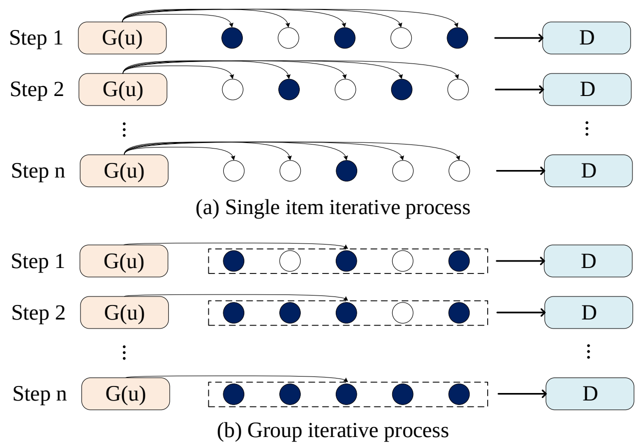


Figure 1: The illustration of sampling products during the training process with 10 items in item pool (including 5 observed items in blue). Subfigure (a) adopts the sampling strategy in IRGAN, at the beginning of the training (Step 1), most of items randomly sampled by generator appear in ground truth item pool, confusing the discriminator since it will be asked to regard an item as both true and fake. Consequently, the sampling results in Step n suffers from serious problem of contradicting labels. Subfigure (b) demonstrates our group-based methods in BiGAN.

tasks.

To avert the limitations of existing GAN-based methods in recommendation tasks mentioned above, in this paper, we propose an innovative bidirectional adversarial framework, namely `BiGAN`. It is a two-fold model including ForwardGAN and BackwardGAN, respectively. ForwardGAN aims to generate a group of possible interacted items instead of a single discrete item index for a particular user. By feeding the produced item groups, the discriminator in ForwardGAN can be well trained and provide proper guidance for the learning of the generator. To illustrate, we take Figure 1 (b) as an example. At the beginning of the learning (Step 1), we can learn the model adequately since all the selected items are used to form one group as the output of the generator. This group will not confuse the discriminator because the two groups' assessments will be distinct even if they have only one item in difference. The same situations also happen in Step 2 and Step n, and thus our constructed group can solve the contradicting labeling problem. BackwardGAN aims to detect an implicit friend (with similar behaviors) depending on the item group that comes from ForwardGAN. An underlying assumption is that a group of items can precisely summarize the preference of a user to some extent. Thus the most reliable friend can be determined by way of the exhaustive analysis of the group. To well train `BiGAN`, BackwardGAN passes the similar user back and makes a comparison with the original user using ForwardGAN, which can discover their possible relationship. Through this training scheme, ForwardGAN can learn remarkable user and item representation with linear space complexity during the bidirectional adversarial training.

We do extensive experiments using four real-world datasets, and demonstrate that our model outperforms other state-of-the-art frameworks in terms of two evaluation metrics (Precision and NDCG). Our main contributions of this paper can be summarized as follows:

- To address the restriction of existing GAN-based recommendation methods, we propose a bidirectional GAN named `BiGAN`, including ForwadGAN and BackwardGAN. It alleviates the limitation of sparse problem by considering implicit user relations. It propagates the product of each GAN forward and backward to learn user and item representation in depth. With bidirectional adversarial training, these two GANs can boost each other iteratively, providing superior ranking accuracy.
- In `BiGAN`, we introduce the ForwardGAN that maps an original user to a group of items to tackle the issue of assigning both real and fake labels to the same items. The BackwardGAN is presented to form a similar user relative to the original user based on the group of items and then passes it back to ForwardGAN. Then a similarity exploration strategy is served to build up excellent user representation because the relationship between users is considered and exploited carefully.
- We evaluate our `BiGAN` on four real-world datasets for the top-N recommendation, showing that our framework can significantly outperform state-of-the-art counterparts in terms of ranking accuracy.

2 Related work.

2.1 Collaborative Filtering. For decades, collaborative filtering (CF) has been widely applied in recommender systems [18, 22]. In a variety of CF techniques, matrix factorization (MF) [23] is the most popular one for implicit feedback. In early research, (Hu et al., 2008) [14] propose a matrix factorization method called WRMF, and adopt a regularized least-square optimization, which corresponds to MLE for a normally distributed random variable. Then (Rendle et al., 2009) [13] derive an MF-based pairwise optimization criterion for personalized ranking named BPR, which is a highly competitive approach for item recommendation. Demonstrating the lack of consideration of negative items in pairwise methods like BPR, (He et al., 2016) [15] introduce eALS to optimize the implicit MF model by assigning various weights on negative items. Besides, (He et al., 2018) [16] find that BPR is not robust and vulnerable to adversarial perturbations on network parameters, and propose APR to remove user bias through adding adversarial perturbations on the representations to construct a better and more robust recommender model.

2.2 GAN-based Recommendation Methods. Recently, GAN-based collaborative filtering methods have gained more attention in recommender systems. `IRGAN` [9] is the first GAN-based recommendation framework, in which the generator G samples a single item and the discriminator D detects its confidence. However, D is not trained properly to discover the difference between true and fake items due to noisy and confusing item labels. To address the aforementioned issues, `PD-GAN` [7] classifies items into k categories according to their genre, and constructs multiple diverse item sets for a specific user by sampling diverse items. This method decreases the probability of assigning contradicting labels to the same items. Nevertheless, it treats categories that users have interacted with equally, fully overlooking the preference of users towards different categories. Different from the above methods, `CFGAN` [8] learns a user-specific scoring vector of all items for each user instead of sampling a single item as `IRGAN`. This strategy can address the problem of contradicting labels because it is less likely for the learned real-valued scoring vectors to be identical to the ground-truth scoring vectors. However, it requires much more space to store the scoring vectors for all users, and may fail to achieve significant ranking accuracy when the item size is large. This is because it is difficult for every user to maintain a promising rating vector given such a lot of parameters to learn at the same time. From these perspectives, `CFGAN` is especially infeasible and space-consuming to deal with large datasets.

3 Our Model

3.1 Notations. For the sake of discussion, we first present some notations. Let $U = \{u_1, u_2, ..., u_m\}$ represent the set of users and $I = \{i_1, i_2, ..., i_n\}$ denote the set of items, where m and n refer to the number of users and items, respectively. For the j-th user $u_j \in U$, we define $S_r^{u_j} = \{i_{r_1}, i_{r_2}, ..., i_{r_p}\}$ $(i_{r_k} \in I)$ as an observed group of items (ground truth), which can be used to portray the preference of u_j, accurately, and p is the size of the group. Then $S_f^{u_j} = \{i_{f_1}, i_{f_2}, ..., i_{f_p}\}$ $(i_{f_k} \in I)$ is the generated (fake) group of items for user u_j. Note that $S_r^{u_j}$ and $S_f^{u_j}$ are of the same size. [1]

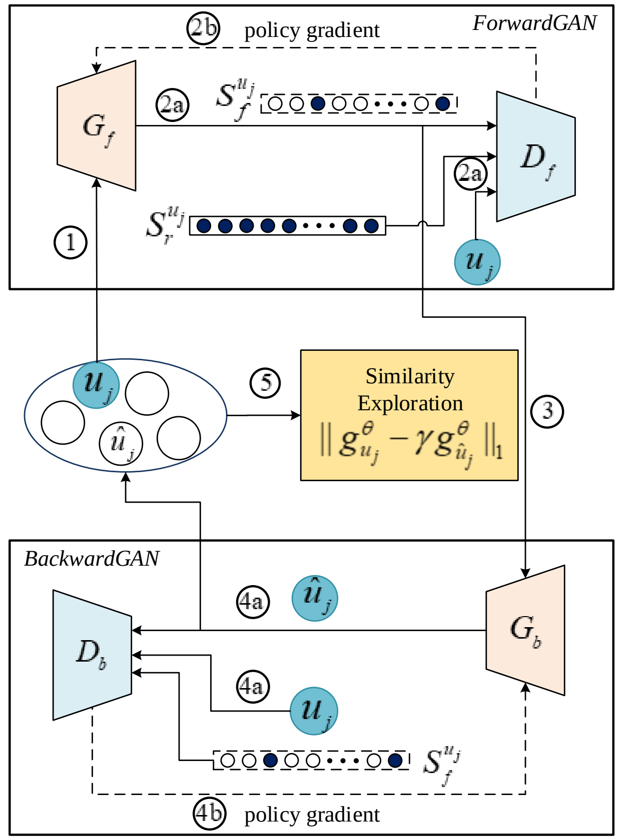


Figure 2: The workflow of our `BiGAN`, which includes ForwardGAN (up, forward) and BackwardGAN (down, backward). For a specific user u_j, user $\hat{u}_j$ is his similar generated user. The numbers in circle illustrate the sequence of our framework operations.

3.2 General Framework. The general framework of `BiGAN` is illustrated in Figure 2. In the ForwardGAN, we first take a real user u_j as input to the generator G_f, as shown in step ①. G_f computes the item sampling distribution by considering the relevant score of all items with u_j and then samples a plausible fake item group $S_f^{u_j}$ accordingly. Next, in step ②ⓐ, the discriminator

[1] It makes sure that the generated group $S_f^{u_j}$ is descriptive and reliable. To illustrate, when $|S_r^{u_j}|$ is 20, if we sample 5 items to form the generated group, it will be problematic because 5 items may not be competent for describing user preference in various aspects comprehensively as 20 items can. However, if $|S_r^{u_j}|$ is designed to be 50, unrelated items may be forced to be included as the model cannot find such a lot of related items, which will further deteriorate the model training. Therefore, we believe that keeping two groups in the same size is reasonable.

(classifier) D_f receives both the fake item set $S_f^{u_j}$ (from G_f) as well as ground-truth item set $S_r^{u_j}$, and attempts to unmistakenly determine whether they are real or not. Gaining effective rewards from D_f by adversarial training (step 2b), G_f will be able to form a group of comprehensive and free-of-contradicting items as similar as the ground truth so as to confuse D_f as much as possible. In BackwardGAN, receiving fake group of items $S_f^{u_j}$ from G_f as input in step 3, the generator G_b will generate a similar user $\hat{u}_j$ that is most likely to interact with items in $S_f^{u_j}$ according to the overall representation of items in the group. In step 4a, we take the real user u_j and the fake (similar) user $\hat{u}_j$ as input to the discriminator D_b, which tries its best to recognize the difference between them. D_b gives a reward back to G_b, which will lead the generator to sample effective similar users, as shown in step 4b. In step 5, we feed the similar user $\hat{u}_j$ backward to ForwardGAN and explore the underlying similarity between two users, which can enlighten the forward GAN to obtain more considerable user representation and enable the whole model to train jointly. In the following section, for the sake of simplicity, we take the user $u_j \in U$ as the original user to illustrate the implementation of our model.

3.3 ForwardGAN.

ForwardGAN is composed of a generator G_f and a discriminator D_f, aiming to construct an item group according to the preference of the user (i.e., $G_f : u_j \to S_f^{u_j}$).

3.3.1 Discriminator D_f in ForwardGAN.

For a specific user $u_j \in U$, the discriminator D_f takes as input both the synthetically generated item set $S_f^{u_j}$ and the ground truth item set $S_r^{u_j}$. To evaluate the reliability and effectiveness of a received group (either true or fake group), we compute the score of each group by matching the characteristic of all items to the user preference, given by:

$$s_\phi(u_j, S_\cdot^{u_j}) = \frac{1}{|S_\cdot^{u_j}|} \sum_{i_k \in |S_\cdot^{u_j}|} ({\mathbf{d}_{u_j}^{\phi}}^\top \mathbf{d}_{i_k}^{\phi} + b_{i_k}^{\phi}), \tag{3.1}$$

where ϕ is the model parameter of D_f. $\mathbf{d}_{u_j}^{\phi}$ and $\mathbf{d}_{i_k}^{\phi}$ represent the embedding of user u_j and item i_k, respectively. $b_{i_k}^{\phi}$ is the bias term of i_k in D_f. Furthermore, $S_\cdot^{u_j}$ represents an item group, either $S_r^{u_j}$ or $S_f^{u_j}$. Finally, we calculate the discriminative possibility $D_\phi(u_j, S_\cdot^{u_j})$, which refers to the possibility that the original group is correctly recognized:

$$D_\phi(u_j, S_\cdot^{u_j}) = \frac{1}{1+\exp(-s_\phi(u_j, S_\cdot^{u_j}))}. \tag{3.2}$$

3.3.2 Generator G_f in ForwardGAN.

We realize our group sampling strategy in G_f to generate a comprehensive item set, where items are most likely to be observed by the user u_j in the future, according to the preference of user u_j. Firstly, we compute the user preference score $s_\theta(u_j, i_k)$ towards an item i_k, given by:

$$s_\theta(u_j, i_k) = {\mathbf{g}_{u_j}^{\theta}}^\top \mathbf{g}_{i_k}^{\theta} + b_{i_k}^{\theta}\ , i_k \in I, \tag{3.3}$$

where θ is the model parameter of G_f. $\mathbf{g}_{u_j}^{\theta}$ is user embedding of u_j and $\mathbf{g}_{i_k}^{\theta}$ denotes item embedding of i_k. $b_{i_k}^{\theta}$ refers to the bias term of i_k in G_f. After that, we can get the sampling probability $p_\theta(i_k|u_j)$ based on preference score by a softmax function, given by:

$$p_\theta(i_k|u_j) = \frac{\exp(s_\theta(u_j, i_k))}{\sum_{i_s \in I} \exp(s_\theta(u_j, i_s))}. \tag{3.4}$$

Now, in order to form the item set $S_f^{u_j}$, we draw $|S_r^{u_j}|$ items according to sampling probability. Finally, after constructing the item group, we can obtain the possibility of sampling such an item group, given by:

$$G_\theta(S_f^{u_j}|u_j) = \frac{1}{|S_f^{u_j}|} \sum_{i_k \in S_f^{u_j}} p_\theta(i_k|u_j). \tag{3.5}$$

3.3.3 Training ForwardGAN.

Discriminator D_f takes as input both the real item set S_r^u from ground truth and the fake item set S_f^u generated by G_f, aiming to strengthen its capacity to classify the given set to the correct category. Therefore, we try to train it as a good classifier by maximizing its objective function, given by:

$$\begin{aligned} L(D_f) = \max_\phi(&\mathbb{E}_{S_r^u \sim p_\phi(S_r^u|u)}[\log D_\phi(u, S_r^u)] \\ &+ \mathbb{E}_{S_f^u \sim G_\theta(S_f^u|u)}[\log(1 - D_\phi(u, S_f^u))]), \end{aligned} \tag{3.6}$$

where conditional probability $p_\phi(S_r^u|u)$ is an underlying real preference distribution over ground truth item set with respect to a specific user. This conditional probability is computed in the way same as Eq.(3.5) through replacing S_f^u with S_r^u. Intuitively, by maximizing $L(D_f)$, the term $D_\phi(u, S_r^u)$ will become as large as possible, however, $D_\phi(u, S_f^u)$ tends to become as small as possible, which conforms with our expectation.

In contrary to discriminator D_f, the generator G_f intends to minimize the log-probability that the discriminator D_f manages to do right classification. In other words, the generator G_f increases the score of its generated item set by shifting its user-item distribution so as to confuse the classification of D_f. Therefore, we express the objective function of G_f as:

$$L(G_f) = \min_\theta \mathbb{E}_{S_f^u \sim G_\theta(S_f^u|u)}[\log(1 - D_\phi(u, S_f^u))]. \tag{3.7}$$

Traditional GAN methods are widely used to cope with continuous data (pixel value in images), and therefore their optimization method cannot be directly applied to ForwardGAN since items are identified by the discrete index. Inspired by IRGAN [9, 10], we bypass the generator differentiation problem by directly performing policy gradient, shown as:

$$\nabla_\theta L(G_f) = \nabla_\theta G_\theta(S_f^u|u)\log(1 - D_\phi(u, S_f^u)). \tag{3.8}$$

The term $\log(1 - D_\phi(u, S_f^u))$ represents the group evaluating reward from D_f and will be passed back to G_f to provide guidance for model update.

3.4 BackwardGAN. BackwardGAN consists of a generator G_b and a discriminator D_b. It is designed to aggregate all peculiarities of items in a given group to explore a similar user with regard to the original user who derives the item group through adversarial training. Afterward, it will, in turn, backward the similar user to ForwardGAN, resulting in better user representation by discovering their possible unrecognized similarity in ForwardGAN.

3.4.1 Discriminator D_b in BackwardGAN. For an original user u_j and his fake item set $S_f^{u_j}$ from ForwardGAN, the discriminator D_b sets a goal to correctly and effectively evaluate whether the reliability of similarity between candidate user u' and original user u_j is sound or not. To conduct this evaluation, we first compute the user-specific relevance score according to the user representation and the overall characteristic of $S_f^{u_j}$, given by:

$$s_\beta(u', S_f^{u_j}) = \frac{1}{|S_f^{u_j}|}\sum_{i_k \in S_f^{u_j}} (\mathbf{d}_{u'}^{\beta\top}\mathbf{d}_{i_k}^{\beta} + b_{i_k}^{\beta}), \tag{3.9}$$

where β is the model parameter of D_b. The embeddings of user u' and item i_k are expressed as $\mathbf{d}_{u'}^{\beta}$ and $\mathbf{d}_{i_k}^{\beta}$, respectively. $b_{i_k}^{\beta}$ is the weight bias term of i_k in D_b. Note that, u' can refer to either an original user or a similar user. Similar to ForwardGAN, we calculate the discriminative score $D_\beta(u', S_f^{u_j})$, which gives an appraisal of the credibility of the similar user, given by:

$$D_\beta(u', S_f^{u_j}) = \frac{1}{1 + \exp(-s_\beta(u', S_f^{u_j}))}. \tag{3.10}$$

3.4.2 Generator G_b in BackwardGAN. Receiving a specific fake item set $S_f^{u_j}$ generated by G_f based on the original user u_j, G_b attempts to generate a similar (fake) user $\hat{u}_j$ ($\hat{u}_j \in U$). To judge the relative similarity between $\hat{u}_j$ and u_j, the group-based similarity score is computed by:

$$s_\alpha(\hat{u}, S_f^{u_j}) = \frac{1}{|S_f^{u_j}|}\sum_{i_k \in S_f^{u_j}} (\mathbf{g}_{\hat{u}}^{\alpha\top}\mathbf{g}_{i_k}^{\alpha} + b_{i_k}^{\alpha}), \tag{3.11}$$

where α is the model parameter of G_b. $\mathbf{g}_{\hat{u}_j}^{\alpha}$ and $\mathbf{g}_{i_k}^{\alpha}$ denote the $\hat{u}_j$'s embedding and i_k's embedding, respectively. $g_{i_k}^{\alpha}$ is the weight bias term of i_k in G_b. Then we compute the distribution $G_\alpha(\hat{u}|S_f^{u_j})$ that a specific user will be sampled by the generator based on the score we have obtained, given by:

$$G_\alpha(\hat{u}|S_f^{u_j}) = \frac{\exp(s_\alpha(\hat{u}, S_f^{u_j}))}{\sum_{\hat{u}_s \in U}\exp(s_\alpha(\hat{u}_s, S_f^{u_j}))}. \tag{3.12}$$

Finally, we generate one similar user $\hat{u}_j$ relative to u_j according to the sampling distribution.

3.4.3 Training BackwardGAN. Being consistent with D_f, D_b is also trained as a superior judge, trying to determine the extent that generated user $\hat{u}_j$ matches the distribution of the item set $S_f^{u_j}$. On the basis of this assumption, we aim to maximize the objective function:

$$\begin{aligned} L(D_b) = \max_\beta(&\mathbb{E}_{u_j \sim p_\beta(u_j|S_f^{u_j})}[\log D_\beta(u_j, S_f^{u_j})] + \\ &\mathbb{E}_{\hat{u}_j \sim G_\alpha(\hat{u}_j|S_f^{u_j})}[\log(1 - D_\beta(\hat{u}_j, S_f^{u_j}))]), \end{aligned} \tag{3.13}$$

where conditional probability $p_\beta(u_j|S_f^{u_j})$ is an underlying true distribution over the original user with respect to a specific group of items.

For a specific group of items, the purpose of G_b is to generate the most analogous fake user $\hat{u}_j$ that best matches the distribution of this group, and thus improves the user representation in ForwardGAN by explicitly pointing out the users' underlying relationship. Therefore, we train G_b by minimizing the following objective function:

$$L(G_b) = \min_\alpha \mathbb{E}_{\hat{u}_j \sim G_\alpha(\hat{u}_j|S_f^{u_j})}[\log(1 - D_\beta(\hat{u}_j, S_f^{u_j}))]. \tag{3.14}$$

In a similar way as ForwardGAN, we also adopt the reinforcement learning method to optimize G_b. The objective function can be reformulated as follows:

$$\begin{aligned} &\nabla_\alpha L(G_b) \\ &\approx \nabla_\alpha log(G_\alpha(\hat{u}_j|S_f^{u_j}))log(1 - D_\beta(\hat{u}_j, S_f^{u_j})), \end{aligned} \tag{3.15}$$

where the reward term $\log(1 - D_\beta(\hat{u}_j, S_f^{u_j}))$ reflects the extent to which generator G_b can deceive D_b. Generally, the higher reward value indicates the similar user is more likely to admire items in the group.

3.4.4 Similarity Exploration.

In order to bridge the gap between two GANs in the backward part and make sure the whole network is trained jointly, we introduce a similarity loss with the original user u_j, and his similar user (be backwarded) $\hat{u}_j$ and implement it to optimize the ForwardGAN. This loss helps fully exploit the characteristic of different users and learn the unexplored shared characteristic of the two users so as to gain a more eminent user representation. We formulate this loss by using L_1 distance with regard to two users, given by:

$$L_{sim}(G_f, G_b) = \mathbb{E}_{u_j \in U}[||\mathbf{g}^{\theta}_{u_j} - \gamma \mathbf{g}^{\theta}_{\hat{u}_j}||_1], \tag{3.16}$$

where $\mathbf{g}^{\theta}_{u_j}$ and $\mathbf{g}^{\theta}_{\hat{u}_j}$ are the user embedding of u_j and $\hat{u}_j(u_j \rightarrow G_f(u_j) \rightarrow G_b(G(u_j)))$, respectively. Furthermore, γ is a noise parameter, ensuring that the embedding values of u_j and $\hat{u}_j$ are distinct since two users' preferences should not be totally the same. In the next subsection, we will integrate this loss into the overall training loss to enhance user representation.

3.5 Overall Objective.

During the training process of BiGAN, ForwardGAN and BackwardGAN take turns to train with linear space complexity (i.e., $O(m+n)$) iteratively and improve the recommendation performance gradually. In each iteration, we first train ForwardGAN and obtain a fake group of items from G_f, and then train BackwardGAN to map the group from G_f to a similar user. The overall objective function of BiGAN is formed as follows:

$$\begin{aligned} L(G_f, D_f, G_b, D_b) = \min_{\theta,\alpha} \max_{\phi,\beta} (&\underbrace{L(D_f) + L(G_f)}_{\text{ForwardGAN}} \\ &+ \underbrace{L(D_b) + L(G_b)}_{\text{BackwardGAN}} + \underbrace{\lambda L_{sim}(G_f, G_b)}_{\text{Similarity Exploration}}), \end{aligned} \tag{3.17}$$

where θ and α refer to parameters of G_f and G_b, respectively. ϕ is parameters of D_f and β denotes those of D_b. Furthermore, λ is a weight parameter, indicating the importance of the similarity exploration strategy.

3.6 Model Evaluation.

After learning all model parameters iteratively, we evaluate our ranking performance based on G_f in ForwardGAN. For each user u, we calculate the estimated score $r_\theta(u, i)$ for each item i, whereby obtaining a ranking list of items:

$$r_\theta(u, i) = {\mathbf{g}^{\theta}_u}^\top \mathbf{g}^{\theta}_i + b^{\theta}_i. \tag{3.18}$$

3.7 Space complexity.

The space complexity of various GAN-based frameworks is presented in Table 1. IRGAN, PD-GAN and BiGAN only need linear space complexity, while CFGAN spends much higher space complexity, which limits its effectiveness in practice.

Table 1: Comparison of different GAN-based models in the space complexity, where m and n denote the size of users and items, respectively.

Methods	Generator	Discriminator
IRGAN	$O(m+n)$	$O(m+n)$
CFGAN	$O(mn)$	$O(mn)$
PD-GAN	$O(m+n)$	$O(m+n)$
BiGAN	$O(m+n)$	$O(m+n)$

Table 2: Detailed statistics of the datasets

Datasets	# Users	# Items	# Ratings	Sparsity
MovieLens 100K	943	1,683	100,000	93.70%
Pinterest	55,187	9,916	1,500,809	99.73%
MovieLens 1M	6,040	3,706	1,000,000	95.53%
Filmtrust	1,508	2,071	35,497	98.86%

4 Experiment

4.1 Experimental Settings

4.1.1 Datasets.

We conduct an extensive experiment on four publicly accessible real-world datasets. The first two datasets are called MovieLens 100K and MovieLens 1M[2], which have been widely used in recommendation tasks. The third one is Filmtrust[3], which is a small dataset crawled from the entire FilmTrust website for movie recommendation by (Guo et al,. 2008) [12]. The last dataset named Pinterest contains implicit feedback provided by another recommendation framework [18]. The details of these four datasets are shown in Table 2. For each dataset, we randomly split 80% of interactions as the training set and the remaining 20% as the testing set.

4.1.2 Comparision methods.

We compare our BiGAN with the following state-of-the-art methods for recommendation[4].

- **BPR** [13] is a classical pairwise ranking model based on implicit feedback, which adopts matrix factorization (MF) to score items for various users.
- **WRMF** [14] assigns different confidence levels to implicit feedback based on the assumption that various feedback may actually be positive or negative. This leads to a factor model that is especially suitable for dealing with implicit feedback.

[2] https://grouplens.org/datasets/movielens/

[3] https://www.librec.net/datasets.html

[4] The source code of PD-GAN is not available and hard to reproduce, thus we do not select it as one of the comparison algorithms.

Table 3: The results of model comparison in MovieLens 100K and Filmtrust. The improvements of our model relative to baseline models with special marks (* and †) are also provided in the table.

Method	MovieLens 100K						Filmtrust					
	Pre@3	Pre@5	Pre@10	NDCG@3	NDCG@5	NDCG@10	Pre@3	Pre@5	Pre@10	NDCG@3	NDCG@5	NDCG@10
BPR	0.3387	0.3124	0.2734	0.3475	0.3276	0.3121	0.3801	0.3681	0.3376	0.4197	0.4422	0.4997
WRMF	0.4916*	0.4749*	0.4418*	0.5003*	0.4889*	0.4711*	0.4164*	0.3872*	0.3367*	0.4653*	0.4842*	0.5288*
eALS	0.4975	0.4671	0.4296	0.5018	0.4814	0.4607	0.4109	0.3722	0.3332	0.4638	0.4710	0.5206
APR	0.3994	0.3699	0.3198	0.4087	0.3873	0.3499	0.4028	0.3824	0.3461	0.4356	0.4533	0.5117
Improvement	**+22.01%**	**+14.78%**	**+8.40%**	**+23.09%**	**+17.69%**	**+12.33%**	**+9.80%**	**+5.53%**	**+4.90%**	**+11.89%**	**+7.50%**	**+8.42%**
IRGAN	0.4072	0.3705	0.3140	0.4222	0.4009	0.3723	0.4092	0.3845	0.3433	0.4352	0.4548	0.5090
CFGAN	0.4924†	0.4484†	0.3993†	0.5045†	0.4718†	0.4378†	0.4201†	0.3910†	0.3466†	0.4666†	0.4933†	0.5228†
Improvement	**+21.81%**	**+21.57.%**	**+19.93%**	**+22.06%**	**+21.96%**	**+20.88%**	**+8.83%**	**+4.50%**	**+1.90%**	**+11.89%**	**+7.50%**	**+9.66%**
`BiGAN`	**0.5998**	**0.5451**	**0.4789**	**0.6158**	**0.5754**	**0.5292**	**0.4572**	**0.4086**	**0.3532**	**0.5221**	**0.5303**	**0.5733**
Average	**+39.82%**	**+37.00%**	**+36.21%**	**+40.27%**	**+37.82%**	**+35.17%**	**+12.57%**	**+7.32%**	**+3.73%**	**+16.82%**	**+13.85%**	**+11.27%**

Table 4: The results of model comparison in MovieLens 1M and Pinterest. The improvements of our model relative to baseline models with special marks (* and †) are also provided in the table. Note that, due to the high space complexity of `CFGAN`, it is difficult to run it on the large dataset Pinterest. Thus, we do not display this result in the table.

Method	MovieLens 1M						Pinterest					
	Pre@3	Pre@5	Pre@10	NDCG@3	NDCG@5	NDCG@10	Pre@3	Pre@5	Pre@10	NDCG@3	NDCG@5	NDCG@10
BPR	0.3384	0.3174	0.2828	0.3442	0.3291	0.3121	0.0062	0.0060	0.0052	0.0130	0.0172	0.0239
WRMF	0.3631	0.3367	0.2973	0.3719	0.3525	0.3335	0.0067	0.0061	0.0053	0.0151	0.0193	0.0265
eALS	0.3501	0.3278	0.2844	0.3569	0.3409	0.3268	0.0065	0.0063	0.0052	0.0146	0.0190	0.0260
APR	0.3646*	0.3385*	0.3001*	0.3727*	0.3536*	0.3337*	0.0066*	0.0062*	0.0054*	0.0143*	0.0189*	0.0261*
Improvement	**+16.39%**	**+13.66%**	**+8.61%**	**+16.67%**	**+14.75%**	**+10.82%**	**+11.94%**	**+11.48%**	**+9.43%**	**+13.25%**	**+12.95%**	**+11.32%**
IRGAN	0.3469	0.3182	0.2675	0.3597	0.3372	0.3040	0.0062†	0.0058†	0.0052†	0.0137†	0.0179†	0.0253†
CFGAN	0.3998†	0.3589†	0.3013†	0.4187†	0.3992†	**0.3781**†	-	-	-	-	-	-
Improvement	**+5.70%**	**+6.63%**	**+7.17%**	**+3.63%**	**+1.33%**	**-2.25%**	**+20.97%**	**+17.24%**	**+11.54%**	**+24.82%**	**+21.79%**	**+16.60%**
`BiGAN`	**0.4226**	**0.3827**	**0.3229**	**0.4339**	**0.4045**	0.3696	**0.0075**	**0.0068**	**0.0058**	**0.0171**	**0.0218**	**0.0295**
Average	**+17.57%**	**+15.16%**	**+11.97%**	**+17.50%**	**+15.33%**	**+12.07%**	**+16.58%**	**+11.93%**	**+10.29%**	**+21.26%**	**+18.31%**	**+15.57%**

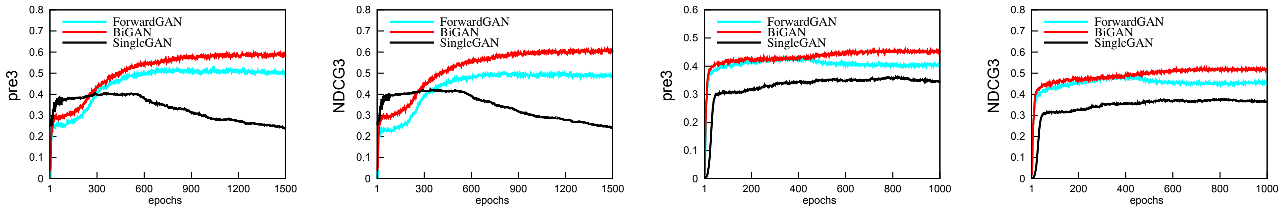
- **eALS** [15] improves MF model by assigning non-uniform weights to unobserved feedback based on item popularity. Besides, assuming that the importance (weight) for unobserved feedback is various in different scenarios, the algorithm utilizes an element-wise Alternating Least Squares (eALS) technique with the missing data whose weight is changing continuously.

- **APR** [16] utilizes adversarial training to improve the robustness of BPR through adding an adversarial regularizer on the model objective function. By making the model resistant to adversarial perturbations, APR enhances the model generalization performance.

- **IRGAN** [9] introduces GAN to sample effective fake items by a learned distribution for model training, and demonstrates GAN's capacity for the success in several IR tasks, including recommendation task.

- **CFGAN** [8] applies a vector-wise adversarial training to solve the problem of the contradicting label that previous GAN-based CF models suffer from by generating a real-valued rating vector for each user. Nonetheless, it is space-consuming since every user needs to maintain an item-sized preference vector.

For the first three baselines, we implementate them using a famous recommendation framework, namely



(a) MovieLens-100k pre@3 (b) MovieLens-100k NDCG@3 (c) Filmtrust pre@3 (d) Filmtrust NDCG@3

Figure 3: Learning curves of our method and its variants on MovieLens 100K and Filmtrust.

Librec[5]. For the others, we run the codes provided by their authors and tune the hyper-parameters carefully.

4.1.3 Parameter Settings. We first empirically set some hyper-parameters that have relatively good performance in practice: parameters in BiGAN are all initialized by a normal distribution $(0, 0.1)$ [19] and the dimension of all embeddings is 20 empirically, the optimizer of two GANs are Adam optimizer [21]. Then we test the learning rate in $\{0.01, 0.0005, 0.0001, 0.00005, 0.00001\}$ and fine-tune in a small step. In addition, for ForwardGAN and BackwardGAN, we take turns to train D with n times and G with just once, where n is tested in $\{1, 5, 10, 15, 20\}$. The weight parameter λ is tuned from 0 to 1 stepping by 0.1, and we evaluate the noise parameter γ in $\{0.8, 0.9, 1.0, 1.1, 1.2\}$.

4.2 Results and Analysis

4.2.1 Comparison with other methods. Table 3 and Table 4 show the ranking results for all methods in comparison within four real-world datasets and the relative performance improvements of BiGAN over the other best methods. Note that, due to the high space complexity of CFGAN, our device (16GB, NVIDIA GeForce RTX 2070) fails to run it on Pinterest successfully. Except that, as shown in tables, our framework almost provides the best ranking accuracy over all the datasets in terms of two metrics.

Comparing GAN-based methods with other traditional methods (without GAN), it is interesting to observe that the former kind of methods does not always show their advantages over the latter ones in terms of ranking performance [17]. For example, in dataset Movielens 100K, WRMF defeats CFGAN in all metrics (excluding Pre@3 and NDCG@3). However, given the fact that our BiGAN can significantly outperform all methods without GAN, which demonstrates that with the correct and effective combination of GAN, a GAN-based model can achieve superior performance by adversarial training.

[5]https://www.librec.net/

For GAN-based models, compared with IRGAN which suffers from contradictory labels problem, CFGAN achieves remarkable improvement over all metrics. These results show that it is effective and significant to avoid attributing two contradicting labels to single items for model training. Furthermore, BiGAN outperforms CFGAN over most of evaluating metrics on three datasets. A possible explanation is proposed to illustrate the point: the active users succeed in seeking similar users in BiGAN to enhance user representation.

4.2.2 Model Ablation. In order to better understand different parts of our BiGAN model, we design two variants of BiGAN:

- **SingleGAN** is exactly the same as IRGAN, whose generator samples a single item at each time. Thus, the discriminator of SingleGAN may receive the contradicting labels for the same item. The reason why we rename IRGAN as SingleGAN is that we try to highlight the single item-based generation in IRGAN.
- **ForwardGAN** removes the BackwardGAN model (including Similarity Exploration) in our BiGAN, and we remain the rest part of it to testify the performance of group-based methods.

The performance curves of variant models are shown in Figure 3, from which we can draw the following insights: (1) The SingleGAN receives the worst results than others, which demonstrates that producing a group of items rather than a single one will have positive influence on model performance by attentively introducing collision-free labels attachment for the same items. (2) Compare the performance of ForwardGAN with our BiGAN, although the two models do not have a significant difference in terms of ranking performance at the beginning, BiGAN will gradually show its considerable advantage over ForwardGAN as the increment of training epochs. This may be due to the fact that by drawing the support from the similarity exploration part, which fully considers and makes use of the correlation among users, we can better fine-tune user representation to some degree.

5 Conclusion and Future Work

In this paper, we present an effective bidirectional GAN-based model, including ForwardGAN and BackwardGAN, to learn superior user and item representation by solving contradicting labels problem and exploiting relationships between users. The ForwardGAN learns to generate and forward a group of user-related items given an original user, which can ensure the effectiveness of discriminator of D_f. BackwardGAN takes the informative item groups as input to produce a similar user and propagates it to ForwardGAN. Moreover, a similarity exploration strategy implemented in ForwardGAN fine-tunes the user representation by exploring the relationship between two users. Our experimental results on four real datasets also confirm that our method is superior to other counterparts. For future work, we will try to introduce some auxiliary information such as social relationships to better search for user similarities, and we will further explore a more effective strategy to exploit similar users.

6 Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (Nos. 61572122, U1736104, 61972078, 61702084), the Fundamental Research Funds for the Central Universities (Nos. N171602003, N181705007), and Huawei Innovation Research Program.

References

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

[2] Gai Li, Zhiqiang Zhang, Liyang Wang, Qiang Chen, and Jin-cai Pan, *One-class collaborative filtering based on rating prediction and ranking prediction*, Knowledge-Based Systems, 2017, pp. 46–54.

[3] Adit Krishnan, Ashish Sharma, Aravind Sankar, and Hari Sundaram, *An Adversarial Approach to Improve Long-Tail Performance in Neural Collaborative Filtering*, CIKM, 2018, pp. 1491–1494.

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

[5] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio, *Generative Adversarial Networks*, NIPS, 2014, pp. 2672–2680.

[6] Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F. Stewart, and Jimeng Sun, *Generating Multi-label Discrete Patient Records using Generative Adversarial Networks*, MLHC, 2017, PP. 286–305.

[7] Qiong Wu, Yong Liu, Chunyan Miao, Binqiang Zhao, Yin Zhao, and Lu Guan, *PD-GAN: Adversarial Learning for Personalized Diversity-Promoting Recommendation*, IJCAI, 2019, pp. 3870–3876.

[8] Dong-Kyu Chae, Jin-Soo Kang, Sang-Wook Kim, and Jung-Tae Lee, *CFGAN: A Generic Collaborative Filtering Framework based on Generative Adversarial Networks*, CIKM, 2018, pp. 137–146.

[9] 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*, SIGIR, 2017, pp. 515–524.

[10] Lantao Yu, Weinan Zhang, Jun Wang and Yong Yu, *SeqGAN: Sequence Generative Adversarial Nets with Policy G radient*,AAAI, 2017, pp. 2852–2858.

[11] Zhi-Dan Zhao, and Mingsheng Shang, *User-Based Collaborative-Filtering Recommendation Algorithms on Hadoop*, WKDD, 2010, pp. 478–481.

[12] Guibing Guo, Jie Zhang, and Neil Yorke-Smith, *A Novel Bayesian Similarity Measure for Recommender Systems*, IJCAI, 2013, PP. 2619–2625.

[13] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme, *BPR: Bayesian Personalized Ranking from Implicit Feedback*, UAI, Montreal, 2009, pp. 452–461.

[14] Yifan Hu, Yehuda Koren, and Chris Volinsky, *Collaborative Filtering for Implicit Feedback Datasets*, ICDM, Pisa, 2008, pp. 263–272.

[15] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua, *Fast Matrix Factorization for Online Recommendation with Implicit Feedback*, SIGIR, 2016, pp. 549–558.

[16] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua, *Adversarial Personalized Ranking for Recommendation*, SIGIR, 2018, pp. 355–364.

[17] Ferrari Dacrema, Maurizio, Cremonesi, Paolo Jannach, and Dietmar, *Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches*, RecSys, 2019, pp. 101–109.

[18] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua, *Neural Collaborative Filtering*, WWW, 2017, pp. 173–182.

[19] Walter Böhm, and Andreas Geyer-Schulz, *Exact Uniform Initialization For Genetic Programming*, FOGA, 1996, pp. 379–407.

[20] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay, *Deep Learning Based Recommender System: A Survey and New Perspectives*, ACM computing surveys, 2019, pp. 5:1–5:38.

[21] Zijun Zhang, *Improved Adam Optimizer for Deep Neural Networks*, IWQoS, 2018, pp. 1–2.

[22] David Goldberg, David Nichols, Brian M Oki, and Douglas Terry. *Using collaborative flitering to weave an information tapestry*, Communications of the ACM, 1992, pp. 35 (12)61–70.

[23] Koren, Yehuda and Bell, Robert and Volinsky, Chris, *Matrix Factorization Techniques for Recommender Systems*, Computer, 2009, pp. 30–37.