Generating Reliable Friends via Adversarial Training to Improve Social Recommendation

Junliang Yu[1], Min Gao[2], Hongzhi Yin[1,*], Jundong Li[3], Chongming Gao[1], and Qinyong Wang[1]

[1]*School of Information Technology and Electrical Engineering, The University of Queensland*
[2]*School of Big Data and Software Engineering, Chongqing University*
[3]*Department of Electrical and Computer Engineering, University of Virginia*
{jl.yu, h.yin1, chongming.gao, qinyong.wang}@uq.edu.au, gaomin@cqu.edu.cn, and jundong@virginia.edu

Abstract—Most of the recent studies of social recommendation assume that people share similar preferences with their friends and the online social relations are helpful in improving traditional recommender systems. However, this assumption is often untenable as the online social networks are quite sparse and a majority of users only have a small number of friends. Besides, explicit friends may not share similar interests because of the randomness in the process of building social networks. Therefore, discovering a number of reliable friends for each user plays an important role in advancing social recommendation. Unlike other studies which focus on extracting valuable explicit social links, our work pays attention to identifying reliable friends in both the observed and unobserved social networks. Concretely, in this paper, we propose an end-to-end social recommendation framework based on Generative Adversarial Nets (GAN). The framework is composed of two blocks: a generator that is used to produce friends that can possibly enhance the social recommendation model, and a discriminator that is responsible for assessing these generated friends and ranking the items according to both the current user and her friends' preferences. With the competition between the generator and the discriminator, our framework can dynamically and adaptively generate reliable friends who can perfectly predict the current user' preference at a specific time. As a result, the sparsity and unreliability problems of explicit social relations can be mitigated and the social recommendation performance is significantly improved. Experimental studies on real-world datasets demonstrate the superiority of our framework and verify the positive effects of the generated reliable friends.

Index Terms—Social Recommendation, Recommender Systems, Adversarial Learning, Generative Adversarial Network

I. INTRODUCTION

The pervasive use of social media services generates a massive amount of data such as news and social feeds, making current Internet users struggle to find relevant information for their own needs. By modeling the historical data of users such as explicit ratings and implicit feedback, recommender systems can capture users' preferences and free them from this information overload problem. Meanwhile, recommender systems create plenty of revenue for e-business companies like Amazon and Alibaba, since users can easily find the content they are really interested in and would like to pay for. However, as users normally only consume a tiny fraction of items among a sea of options, while numerous new items are continuously being produced, traditional recommender

*corresponding author

systems often suffer from the thorny problem of data sparsity and hence fail to satisfy the users.

An effective solution to this dilemma is to transfer knowledge from other fields or platforms and incorporate them into traditional recommender systems. Social relations, particularly, are one type of extremely useful knowledge because people are usually largely influenced by their friends in decision-making [1]. Based on this, it is natural for the academia and industry to explore the application of social networks to recommender systems, subsequently developing a lot of social recommendation methods [2]–[8]. However, according to a recent survey [9], social recommendation is not as successful as expected, and sometimes the incorporation of social connections even degrades the recommendation performance. The failure is attributed to the *diversity* and *unreliability* of social connections. For one thing, a vast majority of existing social recommender systems are based on the simple assumption that all the connected users share similar preferences because of the principle of homophily [10] and hence directly use the explicit social ties for recommendation. But the real situation is rather complex, as the online community is quite different from the offline community in terms of the scale and the possibilities to build connections. Typically, in the online social networks, users can easily expand their friend circles and form different types of relationships such as colleagues, classmates, relatives, among others. Apparently, not all of these social relations have a positive impact on quality-improving for recommendation as most of them may hardly reach a consensus with each other in all the aspects of user preferences. In addition, because of the open nature of social networks, social media users are sometimes mixed with a number of malicious accounts which may pose a threat to accurate preference inference [11]. Hence, it is highly possible that the direct use of explicit social connections will lead to an unsatisfying result.

To tackle this problem, a number of subsequent studies turned attention to extracting salutary relations from explicit social networks [12]–[15]. However, these attempts ignored the fact that social relations are almost as sparse as the user feedback and the filtered explicit relations are inadequate to make a great contribution to improving recommendation quality. For this reason, a better and feasible alternative is to discover *implicit friends* [16], [17] who share similar tastes with the current user but do not have social links with her/him

in the social network. In detail, Ma *et al.* [16] proposed to use rating profiles to search for implicit friends when social networks are not available and Yu *et al.* [17] adopted network embedding techniques to uncover implicit friends for each user. But neither of their studies has an adaptive evaluation mechanism to assess the quality of identified implicit friends. The extracted social links are then fed into the recommendation model based on the simple assumption that they are much more dependable in reflecting users' preferences than original explicit friends even if the searching policy may be independent from the recommendation process. Besides, almost all the social recommendation methods simply model the training as a static process and do not consider the change of similarity or proximity between two users during training. In fact, it is inevitable that after several optimization iterations, some relations will not be conducive to the model and even become noises due to the complexity of the models with multiple tuning parameters. Therefore, approaches which can not only uncover reliable friends but also can dynamically assess these relationships should be developed.

To this end, in this paper we propose a novel social recommendation framework for Top-*N* recommendation which focuses on reliable friends identification with a dynamic evaluation mechanism. Concretely, as the observed social networks are generally very sparse, we firstly identify a few highly reliable friends as *seeded friends* from both the observed and unobserved social networks, who are quite likely to boost the recommendation performance according to existing research [17]. And then a framework based on Generative Adversarial Network (GAN) [18] is developed to dynamically identify reliable social relations under the supervision of the seeded friends and generate high-quality recommendations. More specifically, the framework is composed of two blocks: a generator component and a discriminator component. The generator is responsible for producing a probability distribution that approximates the real distribution of reliable friends, while the discriminator takes charge of ranking the items with BPR [19] as the recommendation module and assesses the quality of generated implicit friends for each user, or in other familiar words, discriminate the reliable friends from undependable friends. With the competition between the generator and the discriminator, friends who can better predict the current user's preference are more likely to be incorporated into the training process while those who may mislead the recommendation model are less likely to be generated. It is noteworthy that this identification process is dynamic, which means the generator can adaptively produce friends for the recommendation model according to the real-time feedback of the discriminator, instead of making the probability distribution of the reliable friends unchanged. As a result, it can bring better recommendation performance.

However, before enjoying the benefits of this built-in dynamic identification process, we have to overcome the discrete item sampling problem in the generator. As each time a friend and one of its items have to be sampled and fed into the recommendation module, our framework cannot be designed like those GANs applied in image generation since the gradient flow would be blocked, which is one of the major issues when GAN is applied in the discrete data domain. Existing IR models such as [20], [21] bypass the generator differentiation problem by utilizing Policy Gradient strategy [22]. Unfortunately, the variance of the estimated gradients scales linearly with the number of items, thus increases the volatility of adversarial training, particularly in the case that millions of items are involved [23], which makes Policy Gradient a sub-optimal solution. Unlike Policy Gradient, the recently proposed Gumbel-Softmax [24] can approximate categorical samples by adding noises sampled from $Gumbel(0, 1)$ to the probability vector produced by the generator, which is known as the reparameterization trick. To enable the framework to be trained via back-propagation, we integrate two Gumbel Softmax layers into our framework to bridge the generator and discriminator and simulate the sampling procedure. Owing to the benefit of end-to-end training, our framework can outperform other recommendation models based on discrete adversarial training.

Overall, the major contributions of this paper are summarized as follows:

- We formally define a social recommendation framework with a built-in dynamic quality-assessing mechanism for social relationships, which, to the best of our knowledge, has not been investigated before.
- We adopt Gumbel Softmax to simulate the friend and item sampling, making our framework trainable with back-propagation, which has been seldom explored in recommender systems.
- We introduce adversarial learning to social recommendation to alleviate the problems of sparsity and unreliability of explicit social relations.
- We conduct experiments on multiple real-world datasets to demonstrate the superiority of the proposed social recommendation framework.

The remainder of this paper is organized as follows. We introduce the related work in Section II. The proposed social recommendation framework is illustrated in Section III. The experimental studies are presented in Section IV. We conclude the whole paper and discuss the future work in Section V.

II. RELATED WORK

In this section, we briefly review related work on two aspects: social recommendation and adversarial learning in recommender systems.

A. Social Recommendation

The early exploration on social recommendation mostly focused on how the explicit social relations can be utilized to improve recommendation performance. A wide range of studies mainly and simply assume that explicitly connected users are supposed to share similar preferences due to the principle of homophily. Among them, SoRec [25] and TrustMF [26] co-factorize the rating matrix and the relation matrix by sharing a common latent space in which purchase and social information

meet with. STE [27] puts forward an ensemble thought that considers user's essential preference as the linear combination of its own explicit preference and those of its friends. Another fusing strategy, social regularization [3] minimizes the gap between the taste of a user and the average taste of its friends through weighted social regularized terms. Later on, some researchers also developed a few models that explain the social influence on users' feedback from different perspectives. A few works [28], [29] noticed that user exposure to items has a great impact on recommendation, and then social connections are used to help capture users' exposure to items instead of preference, which are deemed to be less restrictive. Besides, social information is also leveraged to model the order of items to be recommended. In particular, Zhao *et al.* [30] developed a social Bayesian personalized ranking method that assign higher ranks to items that their friends prefer.

However, as the inferior performance of directly leveraging explicit social information has been reported, many subsequent studies then concentrated on extracting valuable information from social relations and identifying reliable social connections. Guided by the weak tie theory, Wang *et al.* [14] applied the EM algorithm to differentiate strong ties and weak ties from all social ties. Liu *et al.* [6] introduced a new concept, essential preference space, to describe the multiple preferences of users in social recommender systems. In addition, based on the intuition that social tie inherently has various facets indicating multiple trust relationships between users, Tang *et al.* [31] proposed to discern multi-faceted trust in search of experts of different types. Bao *et al.* [32] proposed to decompose the original single-aspect trust information into four general trust aspects and employ the support vector regression to incorporate the multi-facets trust information into the probabilistic matrix factorization model. Enlightened by the success of network embedding techniques, IF-BPR [17] employs a heterogeneous network embedding method [33] to discover reliable unobserved user connections. Besides, a few other studies proposed different trust metrics to statically weigh the quality of social ties by computing and predicting trust scores between users based on interactions [34], [35].

B. Adversarial Training in Recommender Systems

Recently, adversarial learning [18] has achieved great success in various areas such as computer vision and natural language processing. The main idea of adversarial learning is to simulate a minimax game with the generator attempting to imitate the genuine data distribution while the discriminator aiming to differentiate fake examples from the real data. A few pioneering works [20], [21], [36]–[41] have explored the adversarial learning in recommender systems. IRGAN [20] is the first influential IR model constructed based on GAN. This model unifies the generative model and the discriminative model by using the generator to select the informative negative examples and the discriminator to discriminate negative samples from positive samples. Inspired by IRGAN, GraphGAN [21] introduced an alternative of softmax called graph softmax to accelerate training, which can greatly improve the comput-

ing efficiency. Wang *et al.* [37] proposed an adaptive noise sampler to generate adversarial negative samples for neural memory streaming recommender networks, which also boosts the recommendation performance. In addition to the above models focusing on item sampling, Chae *et al.* [39] made the first attempt to directly learn user profiles with GAN instead of sampling items to advance the recommendation model. He *et al.* [40] adopted the thoughts of adversarial examples to recommender systems by adding perturbations to the latent factors of recommendation model, and using adversarial training to lower the risk of over-fitting. Then [41] further extended APR to multimedia recommendation, which also has been shown effective. Besides, [42], [43] investigated a new application of GAN by generating high-level augmented user-item interactions to improve collaborative filtering methods.

III. The Proposed Framework

A. Preliminaries

Firstly, we introduce some important notations used throughout the paper. Let $\mathbf{U}$ denote the user set and $\mathbf{I}$ denotes the item set. m and n are the sizes of the user set and item set, respectively. As our framework is for Top-*N* recommendation, following the convention, we use $\mathbf{R}^{m \times n}$ to represent the interactions between users and items, in which two types of elements are involved: element $r_{u,i} = 1$ indicates that user u like item i and element $r_{u,j} = 0$ indicates that user u dislike item j or item j is unknown to user u.

In our framework, the Bayesian personalized ranking model BPR [19] functions as the recommendation module, which take charges of modeling the order of candidate items. For each user, with the assumption that items appearing in the observed interactions are supposed to be ranked higher than the unobserved ones in the candidate item list, BPR maximizes the margin between the scores of observed items and unobserved ones during training. The assumption of BPR can be formally presented as follows:

$$f : x_{ui} \succeq x_{uj}, r_{ui} = 1, r_{uj} = 0, \tag{1}$$

where f is the model and x_{ui} is the ranking score of item i for user u. Each instance of the training data for BPR is a triple (u, i, j) subjected to the constraint $\{(u, i, j) | r_{ui} = 1, r_{uj} = 0\}$. The loss function of BPR is defined as:

$$\mathcal{L}_{BPR} = -\sum_{(u,i,j)} \log \sigma(x_{ui}(\Phi) - x_{uj}(\Phi)) + \lambda_\Phi ||\Phi||^2, \tag{2}$$

where σ is the sigmoid function, Φ is the model parameters (user latent factors and item latent factors) to be learned and λ_Φ controls the magnitude of Φ to prevent overfitting. By minimizing Eq. 2, BPR can finally obtain good parameters and generate decent personalized recommendation for each user.

B. Motivation

As has been mentioned in Section 1, almost all the social recommendation models hold a static view towards the effects of social relations. From our perspective, the real-time evaluation for relations is indispensable during training. Let
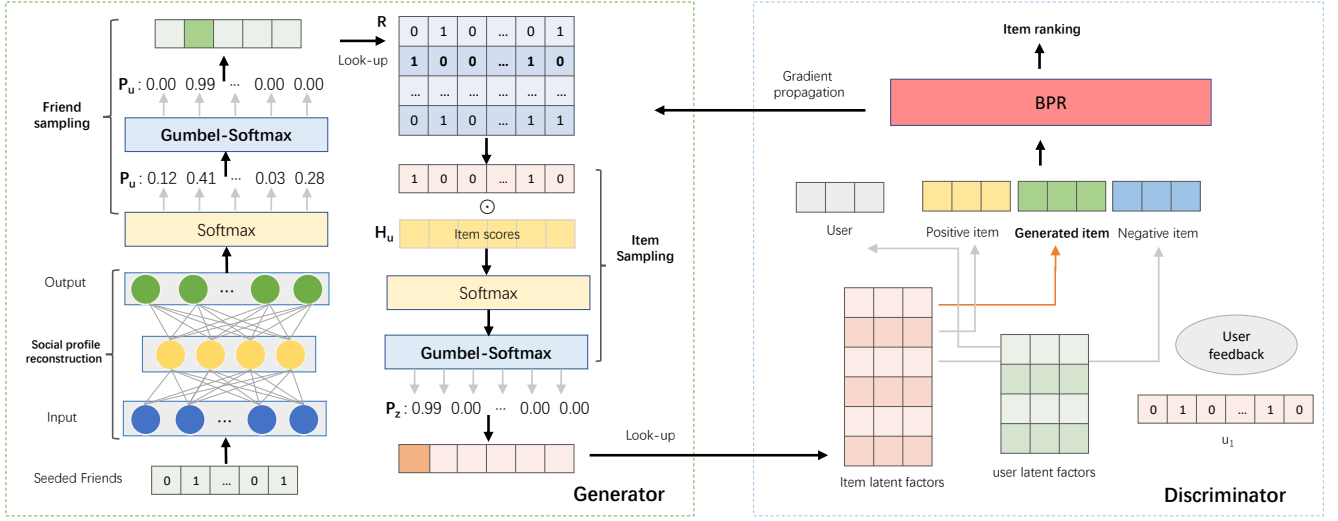
Fig. 1. An overview of the proposed RSGAN framework. The generator outputs a probability distribution over a set of candidate latent friends with the seeded friends as the input. Flowing through two Gumbel-Softmax layers, the probability distribution approximates a one-hot vector, which can be used to simulate the item sampling. The discriminator receives the generated item and outputs a ranked item list as recommendations with user feedback and the generated item as input.

us draw an analogy between the off-line decision making and the training of the social recommendation model. In the real situations, it is common to see that a person consults his friends for advice and then receives different opinions to an item. At first, he may agree with some of his friends but then reaches a consensus with others because of complex psychological activities. Similarly, during the training, social recommendation models are also evolving. A relation initially can be helpful but then may be detrimental to the model due to the complexity of the underlying model with multiple tuning parameters, which resembles the human psychological activities. Therefore, we believe it is completely necessary to build an evaluation mechanism that can assess the real-time effect of relations into the social recommendation model.

C. Model Overview

In this section, we will have an overall view of the proposed **R**eliable **S**ocial recommendation framework , which is based on the **G**enerative **A**dversarial **N**etwork and named **RSGAN**.

It should be noted that RSGAN is designed according to the assumption that users are interested in those items consumed by their reliable friends, which is also the basis of many other social recommendation models. If those items can be identified and fed into the recommendation model to facilitate training, then the recommendation model will be significantly improved. To discover these items, we firstly have to identify the reliable friends for each user, which is the key problem that RSGAN aims to solve. As shown in Fig. 1, working as an end-to-end framework, RSGAN has two major components: the generator (denoted as G_θ) and the discriminator (denoted as D_ϕ) with parameters θ and ϕ respectively. The discriminator takes charge of ranking the candidate items and producing a recommendation list for the user while the generator is responsible for generating a reliable friend and then sampling

an item consumed by this friend which is also likely to be consumed by the current user to enhance the discriminator. Meanwhile, the discriminator punishes the generated friends if items consumed by them are not helpful for advancing the discriminator, and returns gradients to the generator in order to reduce the probabilities of generating such friends.

D. Friend and Item Generation

To guide the generator to predict friends with better quality at any time, we first have to obtain a few of *seeded friends* as the supervisor (ground truth), who are highly probable to be helpful in improving the social recommendation performance. The ways to identify seeded friends can be rather diverse. As shown in [17], heterogeneous network embedding techniques are good at extracting useful links from both observed and unobserved social networks. Following their work, we use the same way to identify a fraction of reliable friends over the social network and the user-item bipartite graph, who are referred to as *perfect friends* in [17], and label them as the seeded friends in our work. However, the number of seeded friends is quite limited, which is not adequate to make a great contribution to improve the quality of recommendation. Hence, the first challenge is that how we can reconstruct the social profile based on the seeded friends in order to have a wide range of friends to be selected. Showing excellent performance in recovering the user rating profile, we believe CDAE [44], which is actually a variant of the fully-connected denoised autoencoders, is also able to reconstruct the user social profile and thus we choose it as the generation block of the generator.

After the seeded friends for each user are collected, we then encode them into binary vectors as the incomplete user social profiles. These incomplete user profiles are fed into the generator as the initial input. By taking the seeded friends as positive examples and the missing ones as negative examples,

CDAE uses cross-entropy loss to optimize the neurons inside and reconstruct the reliable social profiles, and then the layer of softmax outputs the probability distribution over the whole user set $\mathbf{U}$. The distribution can be formally defined as:

$$P_u(v|u, S_u) = \frac{exp(c(v|S_u))}{\sum exp(c(v'|S_u))}, \tag{3}$$

where v is a latent friend of the user u, $c(\cdot)$ is the output of CDAE and S_u denotes the seeded friends of u.

With the possibility distribution of friends known, intuitively, the user with the highest possibility should be sampled as the reliable friend of the user u. However, there are two problems for this kind of choice. Firstly, this user also has high possibility to be chosen next time although the distribution is changing when the optimization proceeds, which may lead to the overfitting of the recommendation model as the items consumed by this user will be sampled a lot of times. Secondly, the discrete sampling procedure is non-differential, which means the training is not end-to-end, resulting in an inferior recommendation performance. A few models [20], [21] tackle the non-differentiation problem by utilizing Policy Gradient strategy [22]. Unfortunately, the Policy Gradient often causes unstable training and slow convergence due to its high variance of the reward, particularly in the case that millions of items are involved, which will further deteriorate the performance of the model [23]. An solution to this issue is to relax the discrete items. Unlike Policy Gradient, the recently proposed Gumbel-Softmax [24] can approximate categorical samples by applying a differentiable procedure, which can be used to optimize large-scale models like recommendation models [38]. To enable the framework to be trained via back-propagation, we integrate two Gumbel-Softmax layers into our framework to simulate the friend and item sampling procedures.

For user u, let $g_u \in \mathbb{R}^m$ denote the noise vector whose elements are randomly drawn from $Gumbel(0, 1)$[1]. The differentiable friend sampling procedure is expressed as follows:

$$v = \frac{\exp\left((\log p_u + g_{\text{noise}})/\tau\right)}{\sum_{i=1}^{n} \exp\left((\log p_{uv'} + g_i)/\tau\right)}, \tag{4}$$

where v is the generated vector that is analogous to a one-hot vector. When the hyper-parameter τ, which is conventionally called temperature, approaches 0, v approximates a one-hot vector, representing the sampled friend. By this reparameterization trick, we have decoupled the sampling procedure from the training process and enabled gradients flowing, but this does not change the behavior of our framework. Meanwhile, the noise brings the sampling procedure some randomness, which helps to avoid the overfitting problem and enhance the model if we can find a balance between the exploitation and exploration of randomness and reliability.

After the reliable friend being sampled, likewise, we sample the item consumed by the friend with Gumbel-Softmax. The difference between the friend sampling and item sampling is that the possibility distribution of items is not generated by

[1] $Gumbel(0, 1)$ can be sampled with $g = -log(-log(\mu))$, where $\mu \sim Uniform(0, 1)$.

the model. Technically, we define a matrix $\mathbf{H} \in \mathbb{R}^{m \times n}$ whose elements are averagely initialized to represent the scores of items, which is going to be updated during training. The item sampling can be formally defined as:

$$z = GumbelSoftmax(v^\top \mathbf{R} \odot \mathbf{H}_u), \tag{5}$$

where z is the analogous one-hot vector that represents the generated item and $\odot$ is the element-wise multiplication used to mask the items that have not been consumed by the sampled friend. Finally, the generated item is delivered to the discriminator to help train the recommendation module. Here we explain that why we do not generate items directly instead of generating a friend at first. In fact, if the number of items is huge, directly generating an item will face a problem that there are millions of candidates, which involve much randomness rather than reliability. By contrast, the number of users is relatively smaller, thus generating a friend under the supervision of the seeded friends and then sampling an item from the limited candidates is more feasible.

E. Adversarial Training

In this paper, we focus on improving the Top-N recommendation and construct the discriminator with the BPR model [19]. To incorporate the generated item from the friends, we follow the principle of Social BPR [30] and impose a social constraint to the discriminator that users tend to assign higher ranks to items that their friends prefer. Hence, the assumption of the recommendation module of the discriminator is formally defined as:

$$\begin{aligned} f : x_{ui} \succeq x_{uz} \succeq x_{uj}, \\ r_{ui} = 1, r_{vz} = 1, r_{uz} = 0, r_{uj} = 0. \end{aligned} \tag{6}$$

With the extra social constraint, the loss function of the discriminator is transformed as follows:

$$\begin{aligned} \mathcal{L}_{D_\phi} = -\sum_{(u,i,z,j)} (\log \sigma(x_{ui}(\Phi) - x_{uz}(\Phi)) \\ + \log \sigma(x_{uz}(\Phi) - x_{uj}(\Phi))) + \lambda_\Phi ||\Phi||^2. \end{aligned} \tag{7}$$

Each time a quad $(u, i, z, j) \in \mathcal{R}$ is input as an instance, in which item i is the positive item consumed the current user, item z is the generated item, and item j is the negative item that has not been consumed by the current user.

Given the loss function of the discriminator, the objective of the generator can be formulated as maximizing the following expectation:

$$\mathcal{L}_{G_\theta} = -\mathbb{E}[\log \sigma(x_{ui} - x_{uz}), z \in \mathbf{R}_v, f \sim P_{G_\theta}(v|u, S_u)]. \tag{8}$$

The objective of generator can be interpreted that the generator tries to produce such friends whose purchased items can obtain a score that can be comparable with the scores of the items purchased by the current user. As the objectives of the generator and discriminator are contradictory, according

to the theory of GAN [18], we have the objective of RSGAN presented as:

$$\mathcal{L}_{G_\theta, D_\phi} = \min_{D_\phi} \max_{G_\theta} -\mathbb{E}[(\log \sigma(x_{ui} - x_{uz})$$
$$+ \log \sigma(x_{uz} - x_{uj}))], z \in R_v, v \sim P_{G_\theta}(f|u, S_u). \quad (9)$$

By playing a minimax game, the parameters of the discriminator can be updated via gradient descent:

$$\Phi \leftarrow \Phi - \alpha \cdot \nabla_\phi \mathcal{L}((f_\phi(i|u) - f_\phi(z|u)$$
$$+ (f_\phi(z|u) - f_\phi(j|u)), \quad (10)$$

while the parameters of the generator are updated via gradient ascent:

$$\Theta \leftarrow \Theta + \alpha \cdot \nabla_\theta \mathcal{L}\left(f_\phi(i|u) - f_\phi(z|u)\right). \quad (11)$$

With the competition between the generator and discriminator, the training process will reach an equilibrium where the generator can produce highly reliable friends and corresponding items while the discriminator can assign good ranks to all the candidate items.

Let us look back to the evaluation mechanism mentioned in Section 3.2. It is obvious that the adversarial training plays such a role. We can see the adversarial training as a type of persistent evolution of the generator and the discriminator. When the generated friend does not provide an item that can contribute more to the recommendation module, then the delivered gradients will penalize the generator and make it updated towards reducing the possibilities of generating such a friend. At the same time, the possibility of producing a better substitution increases and the discriminator will benefit from this update. It should be noted that even the seeded friends are likely to be penalized if they have a negative impact on minimizing the loss of the discriminator at a certain time. As a result, there are no '*extremely safe relations*' since the process can be viewed as a dynamic and adaptive evaluation mechanism which assesses the quality of relations according to the real-time performance. In other words, the evaluation mechanism is integrated into the model training rather than independent of it. With this mechanism, the social recommendation model becomes robust and flexible. Meanwhile, we notice that because of the benefits of applying Gumbel-Softmax, RSGAN becomes fault-tolerant. That is, when the model optimization proceeds and the penalized friend becomes useful to the current user, it still has the possibility to be chosen as Gumbel noises introduce more randomness. Besides, since there are two Gumbel-Softmax layers, penalizing an item only has a limited negative impact on the chances of the friend's other items to be chosen.

In this paper, we concentrate on the task of Top-*N* recommendation. Actually, RSGAN is also suitable for other tasks including rating prediction. The thoughts of social regularization [3] and co-factorization [25] can be easily implemented with RSGAN by re-designing the discriminator and keeping the generator unchanged. The instance in this section just sets an example for other applications. Finally, there is an

---

Algorithm 1: The training process of RSGAN

Input: Seeded friends **S**, user feedback **R**
Output: Recommendation lists and user social profiles.

1 Initialize the generator G_θ and the Discriminator D_ϕ ;
2 Pretrain the CDAE in G_θ with the seeded friends;
3 **for** *each epoch during the training of* G_θ **do**
4   **for** *each user* u **do**
5   Input the seeded friends S_u into the generator;
6   Generates p_u over the whole user set;
7   Feed p_u into the first Gumbel-Softmax layer;
8   Get a one-hot vector representing the friend v;
9   Look-up operation for the items consumed by this friend;
10   Do element product with $\mathbf{H}_u$;
11   Feed the obtained vector into the second Gumbel-Softmax layer;
12   Get a one-hot vector representing the item z;
13   Deliver z to the discriminator;
14   **end**
15 **end**
16 **for** *each epoch during the training of* D_ϕ **do**
17   **for** *each user* u **do**
18   Receive the generated item z from G_θ;
19   Sample a postive item i and a negative item j;
20   Train the BPR model with sampled items;
21   Update D_ϕ based on Eq. 10 and keep G_θ fixed;
22   Update G_θ based on Eq. 11 and keep D_ϕ fixed.
23   **end**
24 **end**

---

unexpected bonus, which is beyond our initial goal, should be mentioned. That is, RSGAN unifies the social recommendation and social link prediction. Because of the interplay of the generator and the discriminator during the training, the CDAE in the generator eventually shows great performance in the task of social link prediction. We will show this bonus in the experiments. The overall process of the training of RSGAN is fully presented in Algorithm 1.

TABLE I
DATASET STATISTICS

Dataset	#Users	#Items	#Feedbacks	#Relations
LastFM	1,892	17,632	92,834	25,434
Douban	2,848	39,586	894,887	35,770
Epinions	18,163	37,325	374,658	287,260

IV. EXPERIMENTS AND RESULTS

To evaluate our proposed framework, experiments are conducted to answer the following research questions: (1). Can RSGAN improve the performance of social recommendation? (2). What are the relevancy and difference between the identified reliable friends and the explicit friends? (3). Can the

TABLE II
PERFORMANCE COMPARISON OF RSGAN AND OTHER METHODS.

| Dataset	Metric	Random	BPR	SBPR	TBPR	NeuMF	IF-BPR	IRGAN	CFGAN	RSGAN	Improv.
LastFM	Precision@10	5.876%	7.598%	8.273%	8.378%	8.114%	9.102%	8.025%	8.968%	**9.475%**	4.098%
	Recall@10	8.622%	11.343%	12.765%	12.965%	12.223%	13.625%	12.477%	13.351%	**14.128%**	3.692%
	NDCG@10	0.08677	0.10857	0.12195	0.12582	0.12331	0.13043	0.12043	0.12975	**0.13509**	3.572%
	Precision@20	4.370%	5.753%	6.254%	6.495%	6.141%	6.825%	6.155%	6.793%	**7.088%**	3.853%
	Recall@20	13.383%	17.824%	18.601%	18.986%	18.674%	20.537%	18.235%	20.232%	**21.091%**	2.700%
	NDCG@20	0.11462	0.13785	0.14664	0.14958	0.14034	0.16252	0.14331	0.16231	**0.16776**	3.224%
Douban	Precision@10	11.776%	13.386%	15.521%	16.071%	16.053%	16.427%	15.144%	15.767%	**17.263%**	5.089%
	Recall@10	3.983%	4.618%	5.0214%	5.325%	5.296%	5.534%	4.878%	5.133%	**6.032%**	8.999%
	NDCG@10	0.12361	0.14857	0.17239	0.17883	0.18047	0.18453	0.16463	0.17647	**0.19469**	5.506%
	Precision@20	9.374%	11.530%	13.005%	13.269%	13.648%	13.884%	12.276%	12.633%	**14.553%**	4.818%
	Recall@20	6.461%	7.533%	8.435%	8.733%	8.690%	8.903%	8.039%	8.371%	**9.887%**	11.052%
	NDCG@20	0.11208	0.13964	0.16116	0.16447	0.16729	0.16931	0.15623	0.16081	**0.18030**	6.491%
Epinions	Precision@10	2.245%	3.152%	3.276%	3.247%	3.213%	3.331%	3.042%	3.260%	**3.477%**	4.383%
	Recall@10	4.983%	6.135%	6.355%	6.418%	6.224%	6.785%	6.033%	6.627%	**7.121%**	4.952%
	NDCG@10	0.04278	0.05423	0.05676	0.05690	0.05538	0.05862	0.05283	0.05787	**0.06082**	3.753%
	Precision@20	1.933%	2.518%	2.626%	2.619%	2.573%	2.635%	2.388%	2.540%	**2.750%**	4.364%
	Recall@20	7.956%	9.729%	10.343%	10.413%	9.918%	10.438%	9.577%	10.310%	**10.993%**	5.317%
	NDCG@20	0.04973	0.06553	0.06896	0.06855	0.06702	0.06986	0.06347	0.06937	**0.07307**	4.595%
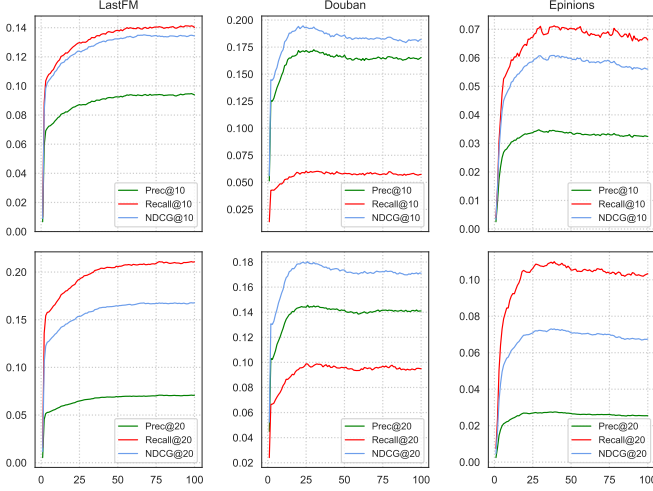
Fig. 2. The learning curve of RSGAN

generator of RSGAN reconstruct the genuine social profiles of the users?

A. Experimental Settings

Datasets. Three widely used social recommendation datasets, LastFM[2], Douban[3], and Epinions[4] are used for experimental evaluations. It should be noted that as the main focus of this paper is to perform Top-N recommendation. So, only the ratings of 4 and 5 in Douban and Epinions are preserved as the user feedback. The statistics of these three datasets are shown in Table 1. For all the datasets, 80% of the data is kept for training, from which 10% is selected for

validation. Specifically, the parameters of baseline methods are determined by their performance on the validation set. Then the experiments are conducted with 5-fold cross validation and the average performances are presented.

Baseline Methods. We compare RSGAN[5] with these popular item ranking methods: BPR [19], SBPR [30], TBPR [13], IF-BPR [17], NeuMF [45], CFGAN [39] and IRGAN [20]. Among them, SBPR and TBPR directly use the explicit relations, and IF-BPR identifies implicit friends with network embedding techniques. These three methods are only suitable for social recommendation. BPR and NeuMF are generic models which are shallow and deep respectively. CFGAN and IRGAN are also based on adversarial learning, and comparing RSGAN with them can show the advantage of utilizing social information. Particularly, to verify that the generated friends are the key point, we randomly select 50 users for each user as their random friends and conduct experiments with only the discriminator available and present the results.

Evaluation Metrics and Configuration. Two relevance-based metrics - *Precision@K* and *Recall@K*, and one ranking-based metric - *NDCG@K* are used to measure the recommendation performance. For all the models, the regularization coefficient λ_Θ is set as 0.001, the batch size is 512, and the dimension of latent factors is 50. For RSGAN, we empirically set the number of sigmoid units in the generator as 200 and the temperature τ in Gumbel-Softmax as 0.2.

B. Recommendation Performance

We first present the the learning curve of RSGAN in Fig. 2. Table 2 and Fig. 3 present the recommendation performance of all the methods on the whole dataset and on the cold-start user set respectively, and we can make the following observations:
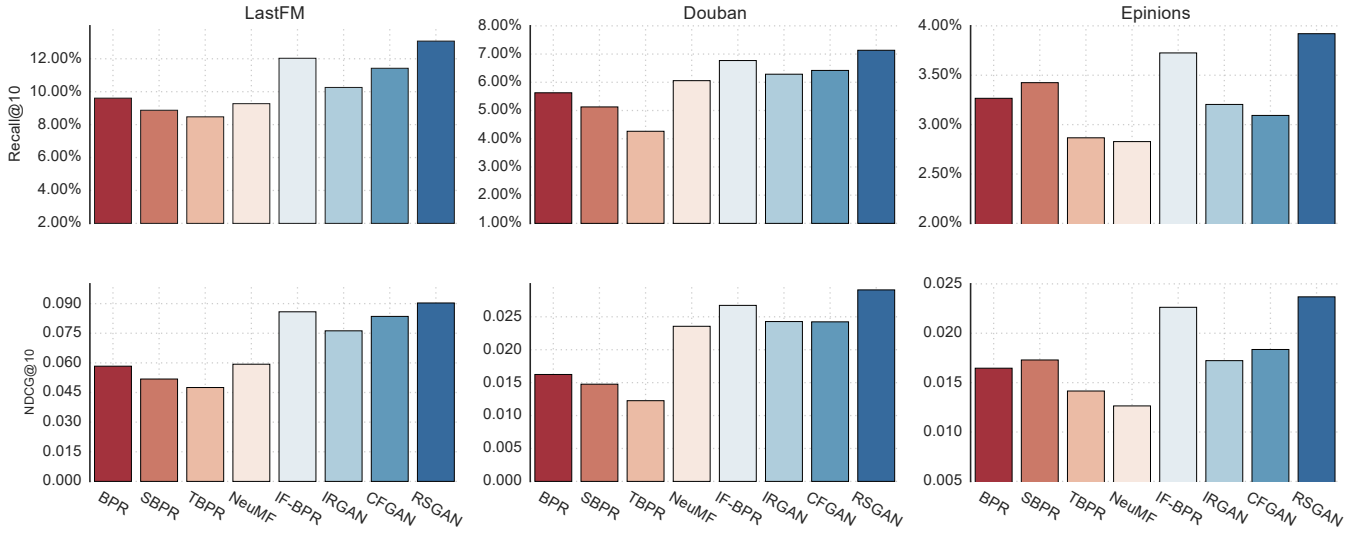

Fig. 3. Evaluation on cold-start users with less than 10 feedback.

1) In all the cases in Table 2, RSGAN outperforms its counterparts on both the relevance and ranking metrics. Specifically, RSGAN achieves good results in terms of the item ranking and the largest relative improvement (calculated by comparing with the second best performance) is more than 11%.

2) The similar results can be observed in the case of recommending for cold-start users as well. RSGAN achieves the best performance while the performance of its main counterparts IRGAN and CFGAN, which is also based on adversarial training, is not very satisfying. We infer that is because IRGAN and CFGAN do not make the most use of the social information and fail to show its capacity in the case of lacking enough user feedback.

3) Overall, in these two tests, methods that directly use explicit social links including SBPR and TBPR do not show satisfying performance, which are in line with the report [9] that the direct use of explicit social relations may have an adverse impact on recommendation performance. Meanwhile, the two implicit friends based methods including IF-BPR and RSGAN show great capacity and beat other methods by a wide margin, which proves that exploiting reliable friends is promising. Besides, the random model with randomly sampled users as friends shows the worst performance in Table 2, which is also a clue to show that it is important to search for reliable friends for social recommendation model. Compared with IF-BPR, RSGAN also gets a decent improvement, we infer it can be attributed to the dynamic evaluation mechanism in RSGAN.

C. Reliable Friends vs Explicit Friends

The purpose of this paper is to identify the reliable friends that can perfectly reflect the current user's preference. It is
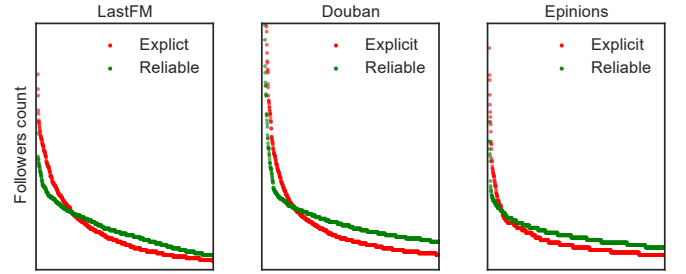

Fig. 4. Follower relationship distribution (1000 users are randomly selected from each dataset).

natural to ask what the relevancy and difference between the identified reliable friends and explicit friends are.

To explore the underlying information, firstly, for each user, we sample another 20 users with the highest probabiliies to be chosen as the reliable friends of corresponding users when RSGAN converges. Then we find that about 60% of seeded friends and about 30% of explicit friends are preserved in the set of finally identified highly reliable friends. If we consider the final set contains the friends who can best predict the corresponding users' preferences, once again, the findings confirm that the explicit social network is noisy for social recommendation. Besides, we use the reliable links to build a reliable social network and compare the one with the explicit social network in terms of the distribution of followers. Generally, the number of followers of each node in the explicit social network is nearly subject to the power-law distribution. Here in Fig. 4. the distribution of followers of these two types of social networks are drawn. In contrast to the follower distribution of the explicit friends, the one of the reliable friends is much more evenly distributed over the whole crowds. More specifically, if most users follow a small fraction of explicit friends, the social recommendation

model is going be less personalized and can not handle the tail users recommendation problem. Besides, users are less likely to meet suprising recommendation originated from the serendipity brought by the diversity of followees, which is like the most-popular recommendation. We believe the follower distribution of reliable links is another reason why RSGAN outperforms other methods.

D. Social link Prediction

Above experimental results have shown the superiority of RSGAN in improving social recommendation. Although RSGAN is not designed to do the task of social link prediction, benefitting from the adversarial training and the user-item interactions involved, the CDAE model integrated into the generator shows great capacity in reconstructing the genuine user social profile, which is beyond our initial expectation and goal. In this section, We compared the strengthened CDAE model with the original CDAE without adversarial learning involved in the training and two widely used link prediction models: DeepWalk [33] and LINE [46] in terms of discovering latent social links in this part.

According to Table 3, we can observe that the strengthened CDAE in RSGAN shows great advantages in predicting the possible social links. We believe this is because the preferences to items have a strong impact on the relation building in social recommender systems, and vice versa, according to the theory of homophily. Particularly, with adversarial training, the interplay between the user-item interactions and user-user interactions can be captured, making the CDAE model fullfil its capacity and meanwhile ehance the framework in return.

TABLE III
PERFORMANCE OF SOCIAL LINK PREDICTION

Dataset	Metric	CDAE	DeepWalk	LINE	RSGAN
LastFM	P@10	4.623%	5.131%	5.206%	**5.587**%
	R@10	14.965%	16.134%	16.396%	**18.271**%
	N@10	0.11187	0.12941	0.13181	**0.14566**
Douban	P@10	3.931%	4.265%	4.133%	**4.519**%
	R@10	12.825%	13.558%	13.407%	**14.361**%
	N@10	0.09876	0.10635	0.10332	**0.11479**
Epinions	P@10	4.664%	4.125%	4.385%	**5.899**%
	R@10	4.206%	3.988%	4.073%	**5.2675**%
	N@10	0.06126	0.05263	0.05457	**0.07362**

V. CONCLUSION

Social recommendation suffers from the sparsity and unreliability problem of the explicit social relations. Inspired by the successful applications of GAN in other areas, in this paper, we present a GAN based friend generation framework, named RSGAN, to improve social recommendation. With the competition between the generator and the discriminator, RSGAN dynamically penalizes the unreliable social relations and adaptively produces reliable friends which can better predict users' preferences. To enable the end-to-end training, we adopt Gumbel-Softmax to relax the discrete sampling, which has been seldom explored in recommender systems. To the best of our knowledge, this is the first work that uses adversarial training to address the unreliability problem of explicit social relations in social recommendation. Experimental analysis on three real-world datasets demonstrates the superiority of RSGAN and verifies the positive effects of the generated reliable implicit friends. In the future, we will extend RSGAN to other social tasks including rating and link prediction.

REFERENCES

[1] J. A. Chevalier and D. Mayzlin, "The effect of word of mouth on sales: Online book reviews," *Journal of marketing research*, vol. 43, no. 3, pp. 345–354, 2006.
[2] M. Jamali and M. Ester, "Trustwalker:a random walk model for combining trust-based and item-based recommendation," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 397–406.
[3] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 2011, pp. 287–296.
[4] G. Guo, J. Zhang, and N. Yorke-Smith, "Trustsvd: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015, pp. 123–129.
[5] W. Fan, Y. Ma, Q. Li, Y. He, Y. E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation," in *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, 2019, pp. 417–426.
[6] C.-Y. Liu, C. Zhou, J. Wu, Y. Hu, and L. Guo, "Social recommendation with an essential preference space." in *AAAI*, 2018.
[7] H. Ma, M. R. Lyu, and I. King, "Learning to recommend with trust and distrust relationships," in *ACM Conference on Recommender Systems*, 2009, pp. 189–196.
[8] L. Wu, P. Sun, Y. Fu, R. Hong, X. Wang, and M. Wang, "A neural influence diffusion model for social recommendation," *CoRR*, vol. abs/1904.10322, 2019.
[9] J. Tang, X. Hu, and H. Liu, "Social recommendation: a review," *Social Network Analysis and Mining*, vol. 3, no. 4, pp. 1113–1133, 2013.
[10] M. Mcpherson, L. Smithlovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annual Review of Sociology*, vol. 27, no. 1, pp. 415–444, 2001.
[11] J. Yu, M. Gao, W. Rong, W. Li, Q. Xiong, and J. Wen, "Hybrid attacks on model-based social recommender systems," *Physica A: Statistical Mechanics and its Applications*, vol. 483, pp. 171–181, 2017.
[12] H. Yin, B. Cui, and Y. Huang, "Finding a wise group of experts in social networks," in *International Conference on Advanced Data Mining and Applications*. Springer, 2011, pp. 381–394.
[13] X. Wang, W. Lu, M. Ester, C. Wang, and C. Chen, "Social recommendation with strong and weak ties," in *ACM International on Conference on Information and Knowledge Management*, 2016, pp. 5–14.
[14] X. Wang, S. C. H. Hoi, M. Ester, J. Bu, and C. Chen, "Learning personalized preference of strong and weak ties for social recommendation," in *The International Conference*, 2017, pp. 1601–1610.
[15] J. Yu, M. Gao, W. Rong, Y. Song, and Q. Xiong, "A social recommender based on factorization and distance metric learning," *IEEE Access*, vol. 5, no. 99, pp. 21 557–21 566, 2017.
[16] H. Ma, "An experimental study on implicit social recommendation," in *International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2013, pp. 73–82.
[17] J. Yu, M. Gao, J. Li, H. Yin, and H. Liu, "Adaptive implicit friends identification over heterogeneous network for social recommendation," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 2018, pp. 357–366.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[19] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 2009, pp. 452–461.

[20] J. Wang, L. Yu, W. Zhang, Y. Gong, Y. Xu, B. Wang, P. Zhang, and D. 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*. ACM, 2017, pp. 515–524.

[21] H. Wang, J. Wang, J. Wang, M. Zhao, W. Zhang, F. Zhang, X. Xie, and M. Guo, "Graphgan: Graph representation learning with generative adversarial nets," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[22] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in Neural Information Processing Systems 12, [NIPS Conference, Denver, Colorado, USA, November 29 - December 4, 1999]*, 1999, pp. 1057–1063.

[23] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, 2014, pp. 1278–1286.

[24] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[25] H. Ma, H. Yang, M. R. Lyu, and I. King, "Sorec: social recommendation using probabilistic matrix factorization," in *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 2008, pp. 931–940.

[26] B. Yang, Y. Lei, J. Liu, and W. Li, "Social collaborative filtering by trust," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 8, pp. 1633–1647, 2017.

[27] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in *International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2009, pp. 203–210.

[28] J. Chen, Y. Feng, M. Ester, S. Zhou, C. Chen, and C. Wang, "Modeling users' exposure with social knowledge influence and consumption influence for recommendation," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 2018, pp. 953–962.

[29] L. Xiao, Z. Min, Z. Yongfeng, L. Yiqun, and M. Shaoping, "Learning and transferring social and item visibilities for personalized recommendation," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 2017, pp. 337–346.

[30] T. Zhao, J. McAuley, and I. King, "Leveraging social connections to improve personalized ranking for collaborative filtering," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM, 2014, pp. 261–270.

[31] J. Tang, H. Gao, and H. Liu, "mtrust: discerning multi-faceted trust in a connected world," in *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 2012, pp. 93–102.

[32] Y. Bao, H. Fang, and J. Zhang, "Leveraging decomposed trust in probabilistic matrix factorization for effective recommendation," in *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*, vol. 350, 2014.

[33] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 701–710.

[34] S. Fazeli, B. Loni, A. Bellogin, H. Drachsler, and P. Sloep, "Implicit vs. explicit trust in social matrix factorization," in *Proceedings of the 8th ACM Conference on Recommender systems*. ACM, 2014, pp. 317–320.

[35] S. M. Taheri, H. Mahyar, M. Firouzi, E. Ghalebi K, R. Grosu, and A. Movaghar, "Extracting implicit social relation for social recommendation techniques in user rating prediction," in *Proceedings of the 26th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, 2017, pp. 1343–1351.

[36] X. Cai, J. Han, and L. Yang, "Generative adversarial network based heterogeneous bibliographic network representation for personalized citation recommendation," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[37] Q. Wang, H. Yin, Z. Hu, D. Lian, H. Wang, and Z. Huang, "Neural memory streaming recommender networks with adversarial training," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 2467–2475.

[38] Z. Sun, B. Wu, Y. Wu, and Y. Ye, "Apl: Adversarial pairwise learning for recommender systems," *Expert Systems with Applications*, vol. 118, pp. 573–584, 2019.

[39] D.-K. Chae, J.-S. Kang, S.-W. Kim, and J.-T. 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*. ACM, 2018, pp. 137–146.

[40] X. He, Z. He, X. Du, and T.-S. Chua, "Adversarial personalized ranking for recommendation," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 2018, pp. 355–364.

[41] J. Tang, X. He, X. Du, F. Yuan, Q. Tian, and T.-S. Chua, "Adversarial training towards robust multimedia recommender system," *arXiv preprint arXiv:1809.07062*, 2018.

[42] Z. Wang, M. Gao, X. Wang, J. Yu, J. Wen, and Q. Xiong, "A minimax game for generative and discriminative sample models for recommendation," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2019, pp. 420–431.

[43] Q. Wang, H. Yin, H. Wang, Q. V. H. Nguyen, Z. Huang, and L. Cui, "Enhancing collaborative filtering with generative augmentation," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2019, pp. 548–556.

[44] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, "Collaborative denoising auto-encoders for top-n recommender systems," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. ACM, 2016, pp. 153–162.

[45] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2017, pp. 173–182.

[46] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th international conference on world wide web*. International World Wide Web Conferences Steering Committee, 2015, pp. 1067–1077.