Mathematical Biosciences and Engineering

MBE, 20(6): 9670–9692.
DOI: 10.3934/mbe.2023424
Received: 05 December 2022
Revised: 05 March 2023
Accepted: 05 March 2023
Published: 22 March 2023

http://www.aimspress.com/journal/MBE

Research article

Recommendation model based on generative adversarial network and social reconstruction

Junhua Gu[1,2], Xu Deng[1], Ningjing Zhang[3] and Suqi Zhang[4,*]

[1] School of Artificial Intelligence and Data Science, Hebei University of Technology, Tianjin 300401, China
[2] Hebei Province Key Laboratory of Big Data Computing (Hebei University of Technology), Tianjin 300401, China
[3] School of Science, Tianjin University of Commerce, Tianjin 300134, China
[4] School of Information Engineering, Tianjin University of Commerce, Tianjin 300134, China

* **Correspondence:** Email: suqizhang@tjcu.edu.cn; Tel: +18602290977; Fax: 022-60438123.

Abstract: Social relations can effectively alleviate the data sparsity problem in recommendation, but how to make effective use of social relations is a difficulty. However, the existing social recommendation models have two deficiencies. First, these models assume that social relations are applicable to various interaction scenarios, which does not match the reality. Second, it is believed that close friends in social space also have similar interests in interactive space and then indiscriminately adopt friends' opinions. To solve the above problems, this paper proposes a recommendation model based on generative adversarial network and social reconstruction (SRGAN). We propose a new adversarial framework to learn interactive data distribution. On the one hand, the generator selects friends who are similar to the user's personal preferences and considers the influence of friends on users from multiple angles to get their opinions. On the other hand, friends' opinions and users' personal preferences are distinguished by the discriminator. Then, the social reconstruction module is introduced to reconstruct the social network and constantly optimize the social relations of users, so that the social neighborhood can assist the recommendation effectively. Finally, the validity of our model is verified by experimental comparison with multiple social recommendation models on four datasets.

Keywords: recommendation algorithm; social recommendation; generative adversarial network; dynamic reconfiguration; graph neural network

1. Introduction

In recent years, with the rapid development of science and technology, the scale of data on the Internet is huge and still growing exponentially. The popularity of portable mobile terminal devices further increases the amount of data on the Internet. Massive data leads to the problem of information overload [1], which makes it difficult for users to pick out the information they need quickly and accurately. In order to improve user experience, recommendation systems have come into being and have been widely used in music recommendation [2], movie recommendation [3] and online shopping [4].

Among many recommendation algorithms, collaborative filtering is the most classical and widely used, but it also has some disadvantages. On the one hand, collaborative filtering algorithms suffer from serious data sparsity and cold start problems. On the other hand, collaborative filtering algorithms cannot capture the high-order nonlinear interaction characteristics of users and items well. Therefore, the recommendation accuracy of classical collaborative filtering algorithms is often not satisfactory.

With the popularity of social platforms, people's decisions are often influenced by their friends, so social relations are often used to alleviate the problem of data sparsity [5,6]. In addition, deep learning technology [7] has achieved great success in many fields in recent years, and recommendation systems also have new opportunities in this context [8,9]. Among them, the generative adversarial network [10] is a representative deep learning technology, which has made great achievements in many fields such as image data generation [11], music composition [12] and medical detection [13]. Since it can learn complex distribution rules of input data and generate consistent distribution with input data, many experts and scholars try to introduce generative adversarial networks into recommendation systems. By learning sparse user-item interaction data, it can capture unknown user preference information and effectively mitigate the impact of data sparsity by generating negative samples with rich information. At present, using the generative adversarial network to complete the recommendation task [14] is a research hotspot in the recommendation field.

At present, the social recommendation models based on the generative adversarial networks first select the friends who are similar to the user's personal interests and preferences and then make use of the friends' opinions to assist the recommendation. However, when selecting friends directly from the original social network, it may select friends with great differences from users' personal interests and preferences, which will affect the recommendation results. Also, when using social relations for recommendation in the generator, it is generally believed that social relations are suitable for various interaction scenarios; but in fact, when users interact with different items, friends have different influences on users. Therefore, this paper proposes a social recommendation model based on generative adversarial network and social reconstruction (SRGAN). This paper makes two main contributions:

1) Based on the generative adversarial network, a new adversarial framework is proposed. On the one hand, the generator is used to pick out the friends who are similar to the user's personal interests and obtain the opinions of friends through the multi-angle influence layer, which includes the overall preference influence and the specific preference influence. On the other hand, the discriminator is used to distinguish the opinions of friends from the personal interests of users. With the competition between the generator and the discriminator, our framework can optimize the social relations of users and could predict the current user's preference effectively.

2) The social reconstruction module is introduced to reconstruct the social network during adversarial training and adaptively generate reliable friends who have the same preference as users, so

that more effective information can be obtained when the social neighborhoods are used to assist recommendation again.

2. Related works

This section will introduce the related work from two aspects: recommendation models based on social relationships and recommendation models based on generative adversarial networks.

In daily life, users not only rely on the introduction and description of the item but also accept the recommendations of acquaintances and friends when shopping. Therefore, the data in the social network affects users' preferences to some extent, which is conducive to improving the quality of the recommendation algorithm. Especially when dealing with data sparsity, a social recommendation algorithm can significantly improve the recommendation performance for sparse users, thus improving user experience [15].

Jamali [16] et al. first designed SocialMF, a recommendation model that incorporated social influence propagation into matrix decomposition. By analyzing the process of trust propagation among users, they believed that the feature vector of user preference was obtained by weighted average of the feature vector of direct neighbor preference. However, assigning the same weight of influence to each friend is not an accurate measure of the impact of social connections on user preferences. In order to solve the above problems, GraphRec [17] used the attention mechanism to distinguish the strength of the first-order social relations of users, assigned different influence weights to each friend according to the strength of the social relations and weighted to get the vector representation of users in the social space. Considering the problem of data sparsity, in order to make full use of social networks to obtain more social information, some research methods extend first-order neighbor aggregation to higher-order neighbor aggregation. Diffnet [18] uses GraphSage [19] to aggregate higher-order neighbors in social space, but Diffnet only considers neighbor aggregation in social space. Therefore, Diffnet++ [20] based on Diffnet, a multi-level graph attention network [21], was adopted. First, it learns how to aggregate the vector representation of different neighbor nodes and then aggregate the vector representations of users in the social space and interactive space. It integrates user preferences in different spaces. Although the use of higher-order neighbor information can effectively alleviate the problem of data sparsity, the above method directly uses the social relations in the social network to assist the recommendation, and users with social relations in the initial social network may not have the same interests and preferences, which will inevitably lead to noise.

In recent years, generative adversarial networks [22] have been widely applied to recommendation models due to their ability to capture the distribution of complex data during training and their strong robustness. By using the generative adversarial network to learn the data distribution of user-item interaction, the user's preference information can be obtained, and the model's ability to learn user-item interaction can be effectively improved. IRGAN [23] was first proposed to apply GAN to the recommendation system and use the generator to learn the distribution of users' preferences for items and generate the index of items that users are most likely to interact with. This can cause an index to be labeled both positively and negatively, confusing the discriminator and sending an error message to the generator. CFGAN [24] and GCGAN [25] fight against the real purchase vector generated by the noise, constantly train the interaction between the user and the item, solve the problem of discrete label confusion and finally make the purchase vector generated by the generator close to the real purchase vector and improve the recommendation accuracy. RSGAN [26] and ESRF [27] models apply confrontation to social recommendation, using friends' information to better capture users' real preference distributions. In RSGAN, the generator samples the items that friends interact with. As the user's favorite items, the discriminator is responsible for distinguishing the items sampled by the generator from the real interactive items, and it makes the items generated by the generator increasingly close to the user's preferences through confrontation training. The generator of ESRF samples a fixed number of friends, and the discriminator is responsible for distinguishing the score of the item based on the user's opinions of friends and their own preferences, where the score of the item is based on the average opinions of the generated friends. Through adversarial training, the friends generated in the generator become more and more reliable, and the friends' opinions are combined to assist the recommendation. However, the above social recommendation model based on generative adversarial network ignores that friends have different influences on users when they interact with different items. In addition, when choosing friends from noisy social networks, friends with large differences from users' interests and preferences will affect the recommendation results.

3. The design of SRGAN model

The principle of GAN is to play a minimax game between generator and discriminator. The focus of the generator is to capture the distribution of real observed data, generate samples that fit the distribution and fool the discriminator. At the same time, the discriminator tries to distinguish whether the input sample is from the generator or not. In this paper, the generator adopts a new way to select friends and generates friends' opinions that coincide with the user's personal interests, while the discriminator is responsible for distinguishing the generated friends' opinions from the user's personal interests. Based on this adversarial framework, this paper designs a new social recommendation model SRGAN based on generative adversarial network and social reconstruction. The model framework, shown in Figure 1, consists of three modules: generator, discriminator and social reconstruction.

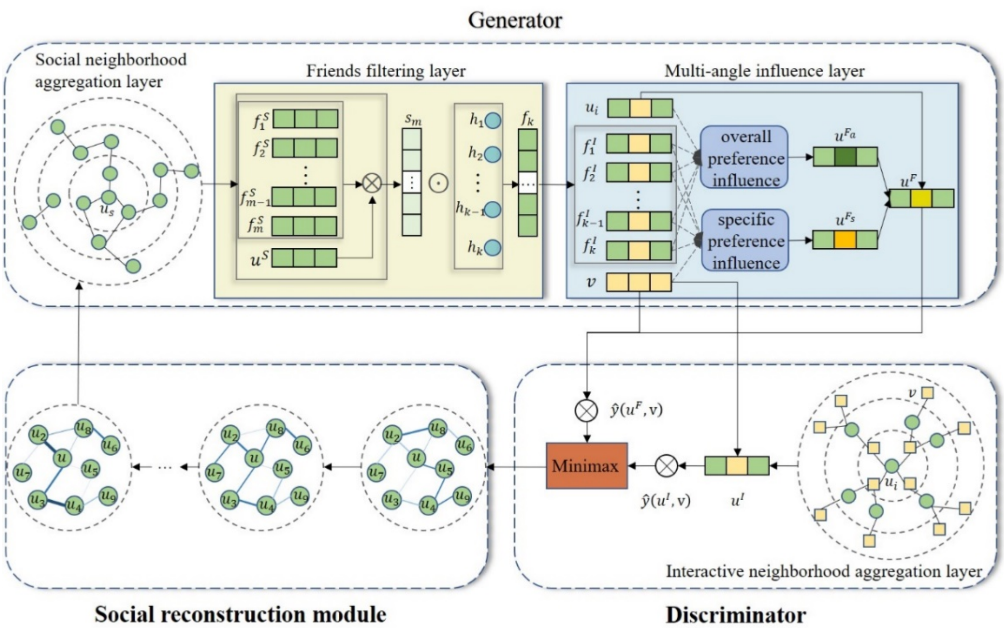


Figure 1. The framework of SRGAN model.

The generator is divided into social neighborhood aggregation layer, friends filtering layer and multi-angle social influence layer. First, the information of friends in the social space is aggregated. Then, pick out the friends $\boldsymbol{f_1^I}, \boldsymbol{f_2^I}, \ldots \boldsymbol{f_k^I}$ whose interests and preferences are similar to the user's. Finally, consider the influence of friends on users from multiple angles to get friends' opinions $\boldsymbol{u^F}$.

Through the user interaction neighborhood aggregation layer, the discriminator uses the user-item interaction graph to aggregate the user's high-order neighbors in the interaction space and obtains the user's personal interest $\boldsymbol{u^I}$. By distinguishing the user's own rating of the item $\hat{\boldsymbol{y}}(\boldsymbol{u^I}, \boldsymbol{v})$ and the user's rating of the item after integrating the friends opinions $\hat{\boldsymbol{y}}(\boldsymbol{u^F}, \boldsymbol{v})$, we can distinguish the friend's opinions and the user's personal interests.

Finally, in the process of confrontation training, the social network is dynamically reconstructed, and the weight of the social relationships around the user is constantly updated according to the feedback of the confrontation loss function, so as to optimize the social relationship between the user and friends. Then, the generator can obtain more effective information when using the social relationship to assist the recommendation again and improve the recommendation performance.

3.1. The generator

The generator is designed to obtain friend opinions similar to the user's personal interests, so that the user can get more accurate recommendation results by combining the friend opinions. It includes social neighborhood aggregation layer, friends filtering layer and multi-angle social influence layer. Take the user *u* as an example. First, aggregate its high-order neighbors in the social relationship graph to obtain the vector $\boldsymbol{u^S}$ in the social space. Then, send the vector $\boldsymbol{u^S}$ to the friends filtering layer in the generator to obtain the friend indication vector $\boldsymbol{f_k}$, and select the friends $\boldsymbol{f_1^I}$、$\boldsymbol{f_2^I}$、$\ldots \boldsymbol{f_k^I}$ with similar preferences to the user. Finally, the influence of the selected friends on the user is divided into overall preference influence $\boldsymbol{u^{F_a}}$ and specific preference influence $\boldsymbol{u^{F_s}}$. By combining the influence of two parts and the initial representation of the user in a bipartite graph, the friends' opinions ($\boldsymbol{u^F}$) are obtained.

3.1.1. Social neighborhood aggregation layer

The number of first-order neighbors of most users in social networks is very limited, and it is difficult to benefit from social relations. Therefore, we use graph convolutional neural network (GCN) to aggregate users' high-order neighbors, enrich users' social relations through high-order relationships and capture the impact of high-order neighbors on users. As shown in Eq (1),

$$u_s^{(L+1)} = \sum_{m \in M_u} \varphi_m f_m^{(L)} \tag{1}$$

where M_u represents a collection of friends with whom the user has a social relationship, φ_m represents the influence of friend *m* on users, f_m is the vector representation of friend *m*, $\boldsymbol{u_s}^{(\boldsymbol{L+1})}$ represents the user representation obtained after *L* layers of convolution in the social graph. By aggregating the high-order neighbor information, we get the representation of each user at different layers: $\boldsymbol{u_s^{(0)}}, \boldsymbol{u_s^{(1)}}, \ldots \boldsymbol{u_s^{(L+1)}}$. Each representation in the set contains the node domain information captured by different layers. In order to make full use of the obtained user embedding, we perform layer combination to obtain the final user representation $\boldsymbol{u^S}$, as shown in Eq (2):

$$u^S = \sum_{l=0}^{L} \frac{1}{L+1} u_s^l \tag{2}$$

Similarly, it is possible to obtain the representation of each user in the social networks, including the representation of the current user's friends $\boldsymbol{f_1^S}, \boldsymbol{f_2^S}, \ldots \boldsymbol{f_m^S}$ in the social space.

3.1.2. Friends filtering layer

First, the degree of preference similarity between user *u* and the friends $\boldsymbol{f_1^S}, \boldsymbol{f_2^S}, \ldots \boldsymbol{f_m^S}$ connected in the social network can be calculated, and the vector representation of the current user can be multiplied with that of other users, as shown in Eq (3):

$$s_m = F_m u^S \tag{3}$$

where $\boldsymbol{F_m}$ is the matrix composed of user's friends $\boldsymbol{f_1^S}, \boldsymbol{f_2^S}, \ldots \boldsymbol{f_m^S}$ in the social space, and $\boldsymbol{s_m}$ is the similarity vector, representing the degree of preference similarity between user and friends.

However, $\boldsymbol{s_m}$ is the degree of similarity between users and their friends in the interactive space, while friends with close relationships in the social space may not have the same purchase preference. In order to make the friends selected from the social space play a guiding role in the user's purchase in the interaction space, the neural network containing *K* neurons is adopted here for adjustment, so as to better learn the degree of similarity between the purchase preferences of users and their friends, so as to select *K* number of friends who are more similar to the purchase preferences of users.

Using $\boldsymbol{s_m}$ and *K* neurons $\boldsymbol{h_1}, \boldsymbol{h_2}, \ldots, \boldsymbol{h_k}$ to do the Hadamard product, *K* friends are selected, as shown in Eq (4):

$$a_k = softmax\left(s_m \odot h_k\right) \tag{4}$$

where $\boldsymbol{\alpha_k}$ is the friends selection indicator vector of user *u*, representing the probability of friends being selected for auxiliary recommendation, that is, friends with *K* bits and similar preferences of users are selected from *m* bits of friends. With every multiplication by a neuron *h*, the probability value of a certain position in the similarity vector $\boldsymbol{s_m}$ will reach the maximum, and eventually the probability value of *K* positions will be too large. Then, the $\boldsymbol{\alpha_k}$ is sent to Gumbel-softmax for friend sampling, as shown in Eq (5):

$$f_k = \frac{\exp\left(\left(\log a_k + g\right)\tau\right)}{\sum_{j=1}^{m}\left(\left(\log a_{kj} + g_j\right)\tau\right)} \tag{5}$$

Gumbel-softmax approximates the classified samples by a differentiable re-parameterization procedure. *g* represents the random noise vector obtained from the Gumbel (0,1) distribution, and $\boldsymbol{\alpha_k}$ via Gumbel-Softmax generates the one-hot-like vector $\boldsymbol{f_k}$, representing the index of friends sampled by the generator for the user. The hyperparameter τ is called temperature according to convention. Choosing a large temperature τ is equivalent to smoothing the probability vector of the social relation and reducing the gap between the probabilities of each label in the original distribution. On the contrary, the closer $\boldsymbol{\tau}$ is to 0 and the closer $\boldsymbol{f_k}$ is to the one-hot vector, the probability of the label with the highest probability in the original distribution will be increased, which represents the discrete friend-sampling process. Because the friends filtering layer contains K neurons, the final generated $\boldsymbol{f_k}$ has K positions of 1, representing the selected friends for auxiliary recommendation.

3.1.3. Multi-angle influence layer

In daily social life, users are more likely to accept suggestions from friends with similar overall preferences in the decision-making process for projects. At the same time, when buying specific items, users usually accept suggestions from friends who really know about the items. For example, if a user needs to buy a computer, he or she will not only consult friends who have similar interests with themselves but also consult friends who know about the computer. This is because friends who have the same preferences as users can put forward their own suggestions on computer color, size and so on, and friends who know electronic products can recommend cost-effective computers. Therefore, in the multi-angle influence layer, the social influence of friends on users is divided into two angles, namely, the overall preference influence and the specific preference influence.

From the angle of the overall preference influence, friends with similar overall preferences of target users should have greater influence. From the angle of the specific preference influence, friends who know more about the specific item should have a greater influence on users. For example, friend A likes watching movies, and friend B likes fitness. When users want to buy fitness products, friend B may have a greater influence on users, and users are more inclined to listen to friend B's opinions.

The overall preference influence part puts the user's personal interest preference and the friend's interest preference vector into the attention network to get the influence weight of each friend on the user. The greater the weight is, the closer the friend's and user's overall preferences are. In the decision-making process of items, users tend to give more consideration to the suggestions of friends with similar habits and preferences. As shown in Eq (6) below,

$$u^{F_a} = \sum_{k \in A_u} \alpha_f f_k^I \tag{6}$$

where A_u is the friends set of u selected in the friends filtering layer, and $\boldsymbol{f_k^I}$ is the vector representation of friends in the interaction space, which can be obtained by the interaction neighborhood aggregation layer of the discriminator module. Since the social space and the interactive space are two different spaces, the friends who have close relationships with users in the social space may not have similar preferences to users in the interactive space. However, we finally recommend items for users in the interactive space, so we use the friend vector of the interactive space to represent $\boldsymbol{f_k^I}$. By aggregating the features of the friends around the user through the attention mechanism, the influence of the friends on the user's overall preference $\boldsymbol{u^{F_a}}$ can be obtained. The attention coefficient α_f can be obtained by Eq (7):

$$\alpha_f = \frac{exp\left(q \bullet \sigma\left(W\left(u_i \oplus f_k^I\right)\right)\right)}{\sum_{f_{k'}^I \in A_u} exp\left(q \bullet \sigma\left(W\left(u_i \oplus f_{k'}^I\right)\right)\right)} \tag{7}$$

where q is the trainable attention parameter, σ is the activation function, $\boldsymbol{W}$ is the weight matrix, $\boldsymbol{u_i}$ is the vector representation of the user in the interaction space, representing the user's personal interest preference, and $\oplus$ is the connection operation.

The influence of specific preference is to consider the differences in preferences between users and friends when interacting with specific items, and the opinions given by friends who know more about specific items are more referential. Therefore, in the process of interaction between users and specific items, according to the characteristics of the item, friends who know this angle are assigned a larger weight to increase the influence of friends who know this type of item. As shown in Eq (8),

$$u^{F_s} = \sum_{k \in A_u} \beta_f f_k^I \tag{8}$$

Equation (8) also aggregates the selected K number of friends through the attention mechanism to obtain the influence of the friend's specific preference $\boldsymbol{u^{F_s}}$ on the user, and the attention coefficient β_f can be obtained from Eq (9):

$$\beta_f = \frac{exp\left(q \bullet \sigma\left(W_1 f_k^I \oplus W_2 v\right)\right)}{\sum_{f_{k'}^I \in A_u} exp\left(q \bullet \sigma\left(W_1 f_{k'}^I \oplus W_2 v\right)\right)} \tag{9}$$

where $\boldsymbol{W_1}$, $\boldsymbol{W_2}$ are trainable parameter matrices, and v is the vector representation of the item in the user-item graph, which can be obtained by the interactive neighborhood aggregation layer. By splicing the item vector v with the friend vector $\boldsymbol{f_k^I}$, we can get the specific preference influence $\boldsymbol{u^{F_s}}$ of friends when interacting with different items.

By integrating the overall preference influence and specific preference influence of friends on users and the vector representation of users themselves, we can obtain the user's representation that integrates the opinions of friends, and we can strengthen the influence of friends who know the item to be recommended and the influence of friends with similar preferences to the target users. As shown in Eq (10),

$$u^F = u_i + u^{F_a} + u^{F_s} \tag{10}$$

3.2. The discriminator

The function of the discriminator is to distinguish the friends' opinions generated by the generator from the user's personal interests. The user's personal interests are obtained by aggregating the items of user interaction through the interactive neighborhood aggregation layer in the discriminator, which is the user's real interest preference.

In the user-item graph, the items express the user's interest characteristics, and the interactive users of the item express the characteristics of the item. However, the interaction data is very sparse, so it is necessary to aggregate the information of the high-order neighbors in the user-item interaction diagram to obtain the feature vector representation of the user and the item. The calculation formula is as follows:

$$u_i^{(L+1)} = \sum_{n \in N_u} \gamma_n v^{(L)} \tag{11}$$

$$v^{(L+1)} = \sum_{n \in N_v} \lambda_n u_i^{(L)} \tag{12}$$

where N_u represents the set of user interaction items, γ_n represents the weight of the items when aggregated, $v^{(L)}$ represents the vector representation of the items at the L layer, and $\boldsymbol{u_i}^{(\boldsymbol{L+1})}$ represents the user representation obtained after convolution at the L layer in the interaction graph. N_v represents the set of user interactions λ_n represents the weight of users when aggregated, $u_i{}^{(L)}$ represents the vector representation of users at the L layer, and $\boldsymbol{v}^{(\boldsymbol{L+1})}$ represents the representation of the item obtained after convolution at the L layer in the user-item graph.

After the representation of users and items in different layers is obtained, the final vector representation of users and items can be obtained by layer combination in the way of Eq (2), as shown in Eqs (13) and (14):

$$u^I = \sum\nolimits_{l=0}^{L} \frac{1}{L+1} u_i^l \tag{13}$$

$$v = \sum\nolimits_{l=0}^{L} \frac{1}{L+1} v^l \tag{14}$$

3.3. Social reconstruction module

Users with explicit social relations in social networks do not necessarily have the same interests and preferences. The term “friends” is broad. The term “friends” in social networks usually includes classmates, colleagues, relatives, etc. Users and different friends have different interests and preferences. Users may not necessarily adopt the opinions given by friends with close social relations, and using it directly will introduce noise. Therefore, how to reduce noise by reconstructing social relations is very important to improve the recommendation effect. In the previous social recommendation model based on the generative adversarial network, the reconstruction of the social relations module is independent of the adversarial training process. RSGAN first pretrains the social reconstruction module separately and then inputs the trained social relations into the generator. However, separating the reconstruction part from the adversarial training cannot better dynamically reconstruct social relations according to the recommended results, so we will dynamically adjust social relations after each adversarial training. What is adjusted here is the weight of the relationships between users and their friends. For those friends who are significantly different from users’ personal interests and preferences, a smaller weight value should be assigned during the aggregation, so as to carry out effective punishment. As shown in Eq (15),

$$U^{(L+1)} = WAU^{(L)} \tag{15}$$

where $\boldsymbol{W}$ is the relationship weight matrix of users and friends, $\boldsymbol{A}$ is the adjacency matrix of nodes in the user social relationship graph, $\boldsymbol{U}$ is the embedded vector matrix of nodes in the user social relationship graph, and L is the number of layers in the convolution. $\boldsymbol{U^{(L)}}$ is the matrix representation of users at the layer L. By adjusting the parameter matrix W, the user's higher-order neighbors in the social space are aggregated again, and the new vector representation of the user in the social space is obtained, which is sent to the generator as the basis for the next confrontation training.

After each adversarial training, $\boldsymbol{W}$ will be dynamically updated, and the social neighborhood aggregation layer in the next generator will aggregate neighborhood friends according to the previous weight. As the adversarial training proceeds, the capabilities of the generator and the discriminator reach a relative balance. The generator can reconstruct a more reliable social network. Friends with similar personal interests and preferences of the user are assigned a greater weight, reducing the noise caused by friends with large differences in preferences. The friends finally selected from the social space are of guiding significance to the user's purchase.

3.4. Adversarial training

In order to obtain better user and item representation, we used a Bayesian personalized ranking (BPR) loss function to pre-train the discriminator before the adversarial training, as shown in Eq (16):

$$L = \sum_{(u,i,j)\in O} -\log\sigma\left(\hat{y}_{u,i}(\Phi) - \hat{y}_{u,j}(\Phi)\right) \tag{16}$$

where Φ represents the parameter of the discriminator module, and σ is the sigmoid function. The triples of the current user u, the item i that u interacts with, and the unknown item j sampled from the observed data are fed back to the model each time.

On the one hand, the friends generated by the generator are similar to the user's preferences and will be interested in the items that the user has highly rated. Therefore, the friends' opinions should be similar to the user's personal interests. On the other hand, when users make purchase decisions, their own interests and preferences should be more critical than their friends' opinions. They will put their interests first, and there will be a gap between their own ratings of the item and those of the item taking into account their friends' opinions. Based on the game of the above frameworks, the goal of the generator is to generate friends whose personal interests are similar to those of the user. Therefore, the difference between the friends' opinions and the user's personal interests should be minimized. The loss function is shown in Eq (17):

$$L_{G_\theta} = \max_{G_\theta} - log\sigma\left(\hat{y}\left(u^I, v\right) - \hat{y}\left(u^F, v\right)\right) \tag{17}$$

where G_θ represents the parameters to be trained in the generator, σ is the sigmoid function, $\hat{\boldsymbol{y}}(\boldsymbol{u^I}, \boldsymbol{v})$ is the user's own rating of the item, and $\hat{\boldsymbol{y}}(\boldsymbol{u^F}, \boldsymbol{v})$ is the user's rating on the item after integrating the friend's opinions. By fixing the parameter of D and maximizing the loss, G evolves towards generating neighbors that can narrow the gap $\hat{\boldsymbol{y}}(\boldsymbol{u^I}, \boldsymbol{v}) - \hat{\boldsymbol{y}}(\boldsymbol{u^F}, \boldsymbol{v})$.

The objective of the discriminator is to distinguish the friends' opinions and the user's personal interests. It is believed that there is a gap between the two preferences, so the score gap should be maximized. The loss function is shown in Eq (18):

$$L_{D_\varphi} = \min_{D_\varphi} - log\sigma\left(\hat{y}\left(u^I, v\right) - \hat{y}\left(u^F, v\right)\right) \tag{18}$$

where D_φ represents the parameters to be trained in the discriminator. By fixing the parameter of G and minimizing the above loss, D is optimized towards recognizing the generated neighbor and making the gap larger. With the competition between G and D, the optimization will eventually reach an equilibrium where the framework shows the best performance.

The training process of the SRGAN model is as follows:

Table 1. The training process of SRGAN model.

Algorithm SRGAN Adversarial Training Algorithm

Input: social relationship matrix $\boldsymbol{S}$, interaction matrix $\boldsymbol{Y}$

Output: trainable parameters of generator G_θ and discriminator D_φ

1 initialization of generator G_θ and discriminator D_φ

2 pre-training of discriminator based on Eq (16)

3 **for each epoch do**

4 **for each batch of G do**

5 transfer social relationship matrix **S** to social neighborhood aggregation layer, and the representation of each user in the social space can be obtained by Eqs (1) and (2), Thus, the representation of the current user and other friends in the social space is obtained: $\boldsymbol{u^S}, \boldsymbol{f_1^S}, \boldsymbol{f_2^S}, \ldots \boldsymbol{f_m^S}$

6 select $\boldsymbol{K}$ friends through Eq (3) to (5), and obtain the interaction space representation of friends $\boldsymbol{f_1^I}, \boldsymbol{f_2^I}, \ldots$ from the interaction space

7 get the friend's opinion $\boldsymbol{u^F}$ through Eq (6) to (10)

8 **end**

9 **for each batch of D do**

10 get the vector representations of users and items $\boldsymbol{u^I}, \boldsymbol{v}$ respectively in the user-item graph through Eq (11) to (14)

11 calculate the user's own score on the item $\hat{\boldsymbol{y}}(\boldsymbol{u^F}, \boldsymbol{v})$

12 calculate the user's own score on the item $\hat{\boldsymbol{y}}(\boldsymbol{u^I}, \boldsymbol{v})$

13 calculate the generator loss according to Eq (17), update the parameter $\boldsymbol{L_{G_\theta}}$

14 calculate the discriminator loss according to Eq (18), update the parameter $\boldsymbol{L_{D_\varphi}}$

15 **end**

16 reconstruct the social network according to Eq (15), update the weight matrix $\boldsymbol{W}$

17 **end**

4. Experiments

In order to verify the performance of the proposed SRGAN recommendation model on Top-k recommendation and score prediction tasks, we conducted experiments on four datasets (Last.FM, Douban, Ciao, Epinions) and compared its performance with the mainstream recommendation models.

4.1. Datasets and evaluation metrics

We tested the performance of the model SRGAN on four real datasets, Last.FM, Douban, Ciao and Epinions. Among them, Last.FM recorded the number of times users listened to the artist's music and the user's social information. Douban includes user ratings of movies and social information among users. Ciao comes from an online social platform that includes user ratings of the items they've purchased and social connections between users. Epinions dataset comes from online social platforms where people can review products, includes information about user ratings of items and social information between users. In this paper, the dataset is randomly divided into the training set and the test set in a ratio of 8:2. The specific statistical results of the dataset are shown in Table 2.

Table 2. Details of the datasets.

Datasets	Last.FM	Douban	Ciao	Epinions
users	1892	2848	7375	40,163
items	17,632	39,586	105,114	139,738
interactions	92,834	894,887	284,086	664,824
relations	25,434	35,770	111,781	442,980

To evaluate the performance of the model, *Precision@ k*, *Recall@ k*, normalized cumulative loss gain (*NDCG@ k*), Mean Absolute Error (MAE) and Root Mean square Error (RMSE) were used for evaluation. In the top-K recommendation task, *k* is set as 10 to rank all candidate items, and the Top 10 items in the recommendation list are selected for evaluation.

Precision represents the proportion of all predicted positive samples containing real positive samples. The definition is as follows:

$$Precision = \frac{TP}{TP + TN} \tag{19}$$

where True Positive (TP) indicates that a positive sample is predicted to be positive, and True Negative (TN) indicates that a Negative sample is predicted to be positive. The higher the accuracy rate is, the higher the recommendation accuracy; otherwise, the recommendation accuracy.

Recall represents the proportion of true positive samples that are predicted to be positive. The definition is as follows:

$$Recall = \frac{TP}{TP + FN} \tag{20}$$

where False Negative (FN) indicates that a positive sample is predicted to be negative.

The definition is as follows: *NDCG* represents the comprehensive evaluation score of correlation and ranking of the items in the test set in the Top-K recommendation list. The larger the *NDCG* value is, the better the ranking result is. The definition is as follows:

$$NDCG = \frac{DGG}{IDGG} \tag{21}$$

$$DCG = \sum_{i=1}^{|REL|} \frac{2^{rel_i} - 1}{\log_2(i+1)} \tag{22}$$

where the $|REL|$ says the results according to the correlation from big to small order, which is in accordance with the optimal way to sort the results. rel_i represents the correlation score of item i. DCG (discounted cumulative gain) considers both correlation and sequential factors to calculate the score of the item in the recommendation list of user u. IDCG (ideal discounted cumulative gain) is the result of DCG normalization.

MAE is the mean of the error between the predicted score and the true score, which reflects the similarity between the predicted score and the true score. The definition is as follows:

$$MAE = \frac{\sum_{(u,i)\in R_{test}} \left| r_{ui} - r_{ui}^{'} \right|}{\left| R_{test} \right|} \tag{23}$$

where $|R_{test}|$ denotes the number of test set users of item evaluation, r_{ui} is the actual user rating of the item, and r'_{ui} is the rating predicted by the model.

RMSE is the square root of the ratio of the squared error between the predicted score and the true score to the number of observations n. The definition is as follows:

$$RMSE = \sqrt{\frac{\sum_{(u,i)\in R_{test}} \left(r_{ui} - r_{ui}^{'} \right)^2}{\left| R_{test} \right|}} \tag{24}$$

Higher *Precision*, *Recall* and *NDCG* values indicate better recommendation performance. *MAE* and *RMSE* reflect the degree of similarity between the predicted score and the real score, and the smaller the value is, the higher the accuracy of the recommendation.

Table 3. Parameters' Settings.

Parameters	Last.FM	Douban	Ciao	Epinions
epoch	30	30	70	120
lr	0.001	0.001	0.002	0.005
batch_size	512	512	2000	2000
d	50	50	50	50
K	30	30	40	50
τ	0.15	0.20	0.20	0.20
ω	0.2	0.2	0.2	0.2
L	3	2	2	2

In the experiment, the default settings of the four datasets are shown in Table 3, where epoch is the training times of the model, *lr* is the learning rate, *batch_size* is the batch size, *d* is the vector embedding dimension of users and items, *K* is the number of selected friends, τ is the temperature coefficient of controlling the generation of neighborhood friends, ω is the coefficient of adjusting the resistance training. *L* is the number of propagation layers of GCN.

4.2. Baseline models

To evaluate the performance of ESRF, we compare it with the following methods: SBPR [28], SoMA [29], CFGAN [24], GCGAN [25], DiffNet++ [20], Light_NGSR [30]，GNN-DSR [31], RSGAN [26] and ESRF. Among them, SBRP, SoMA, DiffNet++, Light_NGSR, GNN-DSR, RSGAN, ESRF are social recommendations. SBPR and SoMA are Bayesian-based social recommendation models. DiffNet++, Light_NGSR and GNN-DSR are based on graph convolutional neural networks. CFGAN, GCGAN, RSGAN, ESRF are recommendation models based on generative adversarial networks. CFGAN and GCGAN are collaborative filtering recommendation models based on generative adversarial network. RSGAN and ESRF both combine social relations and generative adversarial network.

1) SBPR. For the first time, social relations were added to Bayesian Personalized Ranking, which suggested that users prefer the items that their friends like, rather than the items that they have negative feedback or no feedback on.

2) SoMA. It is a social recommendation model using implicit social structure through Bayesian generation model.

3) CFGAN. It is a collaborative filtering recommendation model based on the generation of adversarial network. The generator generates the user's purchase vector, and the discriminator is used to distinguish whether the input vector is real data or the "forged" vector of the generator.

4) GCGAN. On the basis of CFGAN, the discriminator distinguishes whether the input vector is the real data or the purchase vector generated by the generator through the graph convolution network.

5) DiffNet++. It is a social recommendation model based on graph convolution network, which aggregates high-order neighbors in social relationship graph and item interaction graph, distinguishes the influence of neighbors on users by attention mechanism and obtains the prediction score from the inner product of user vector and item vector.

6) Light_NGSR. It is a social recommendation model that uses a lightweight GNN framework, only retains the neighborhood aggregation component and gives up the feature transformation and nonlinear activation component. It aggregates the higher-order neighbor information of user-item interaction graph and social network graph.

7) GNN-DSR. It is a social recommendation model that considers both dynamic and static representations of users and items and incorporates their relational influence. It models the short-term dynamic and long-term static interactional representations of the user's interest and the item's attraction, respectively.

8) RSGAN. It is a recommendation model based on social relations and generative adversarial networks. The generator generates the friend interaction items as the user's favorite items, and the discriminator is used to distinguish the friend interaction items from the user's real favorite items.

9) ESRF. It is also a recommendation model based on social relations and generative adversarial networks. The generator generates friends with similar preferences to the user, and the discriminator

distinguishes the user preference that integrates the friend information from the average friend preference.

4.3. Performance comparison

Comparative experiments were conducted using Precision@10, Recall@10, NDCG@10, MAE and RMSE as metrics. Because the model in this paper makes use of both social relations and generative adversarial networks, we compare it with the social recommendation model SBRP, RSGAN, DiffNet++, ESRF, SoMA, Light_NGSR, GNN-DSR. We use the model based on generative adversarial networks GCGAN, CFGAN, RSGAN, ESRF to explore the performance of the model. RSGAN and ESRF contain both social relationships and generative adversarial networks.

4.3.1. The experimental results of social recommendation model

The experimental results of the recommendation model using social relations on the Last.FM, Douban, Ciao and Epinions datasets are shown in Tables 4–6. The optimal values of the evaluation index in all the tables are highlighted in bold, and the sub-optimal values are underlined below.

In order to better explore the performance of our proposed model in the field of social recommendation, we will divide it into two groups for comparison with the current baseline model of social recommendation: one group for comparison of ranking metrics and the other for comparison of rating metrics. The experimental results are shown in Tables 4 and 5, respectively.

Table 4. The experimental results of social recommendation model.

Datasets	Metric	SBPR	RSGAN	DiffNet++	ESRF	SRGAN
Last.FM	Precision@10	0.1508	0.1535	0.1563	<u>0.1636</u>	**0.1656**
	Recall@10	0.1537	0.1562	0.1584	<u>0.1653</u>	**0.1686**
	NDCG@10	0.1824	0.1910	0.1949	<u>0.2004</u>	**0.2039**
Douban	Precision@10	0.1549	0.1726	0.1737	<u>0.1823</u>	**0.1847**
	Recall@10	0.0502	0.0603	0.0613	<u>0.0654</u>	**0.0670**
	NDCG@10	0.1834	0.1926	0.1954	<u>0.2103</u>	**0.2132**

By observing the experimental results, it can be found that, compared with the baseline models, the SRGAN proposed in this paper obtains optimal values in each index of the two datasets. Compared with the second-best value of each index, SRGAN has an increase of 1 to 2 percentage points in each index.

Table 5. The experimental results of social recommendation model.

Datasets	Metric	SoMA	Light_NGSR	GNN-DSR	SRGAN
Ciao	MAE	0.7859	0.7365	**0.6978**	<u>0.7012</u>
	RMSE	0.9988	0.9736	<u>0.9444</u>	**0.9408**
Epinions	MAE	1.0506	0.8353	<u>0.8016</u>	**0.7956**
	RMSE	1.1890	1.0846	<u>1.0579</u>	**1.0475**

It can be seen from the experimental results in Table 5 that the results of GNN-DSR on MAE are better than the model SRGAN proposed on the Ciao dataset. This is because GNN-DSR considers the possible changes in the attraction of items over time and models users' short-term interest preferences through the sequence of items that users interacted with. The relatively small dataset Ciao can better capture the short-term interest preference of users and improve the accuracy of recommendation. Although our model SRGAN does not use the interaction sequence of users to capture short-term interest preferences, we make use of the opinions of friends with similar preferences of users and capture user preferences by overall preferences and specific preferences, which can predict user preferences stably and improve the recommendation performance. Therefore, the RMSE values of our model on the Ciao dataset and the MAE and RMSE values on the Epinions dataset are better than the baseline models. The results of the experiment show that the overall performance of our model is optimal. Through further analysis of the experimental results, we can draw the following conclusions:

1) Compared with SBRP, which adds social relationship to BPR for the first time, the model with GCN has better performance in Diffnet++, ESRF and SRGAN, because SBRP only considers the first-order neighbor information of user node, while the model based on GCN can aggregate the information of higher-order neighbors. There is effective mining of user-item interaction information and association information in social relations, so as to obtain rich user vector representation.

2) Compared with Diffnet++ using GCN, ESRF and SRGAN are better than Diffnet++ in each evaluation index. These three models all use GCN to aggregate high-order neighbor information, but in addition, ESRF and SRGAN also integrate the generative adversarial network, which shows that integrating friend opinions into the generative adversarial network helps improve the quality of social recommendation.

3) Compared with SoMA, Light_NGSR and GNN-DSR, which only use social relations, SRGAN is almost superior to these baseline models in the two real datasets, indicating that the application of generative adversarial network in the design recommendation is beneficial to improve the accuracy of the model and reduce the scoring error.

4.3.2. The experimental results of recommendation model based on generative adversarial network

In order to verify that SRGAN's thought of opposing friends' opinions against users' personal interests and preferences is conducive to improving the accuracy of recommendation results, we compared it with other baseline models based on generating adversarial networks. The experimental results on Last.FM and Douban datasets are shown in Table 6.

The following conclusions can be drawn by observing the experimental results in Table 6:

1) In the GAN-based model, RSGAN, ESRF and SRGAN have more advantages than CFGAN and GCGAN in various indicators. This is because CFGAN and GCGAN only make use of user-item interaction information, while the other three methods all make use of social information to assist recommendation, which proves that effective use of social data is conducive to improving recommendation performance.

2) SRGAN is superior to the five comparison algorithms in all indicators. Compared with the benchmark model, SRGAN has more obvious performance advantages, which can effectively solve the noise problem caused by social information and improve the accuracy of recommendation. However, SRGAN proposed a new adversarial framework, social reconstruction module and multi-angle influence layer. The real influence of each part on model experimental results cannot be accurately known from Table 6. Therefore, the influence of each module on the experimental results

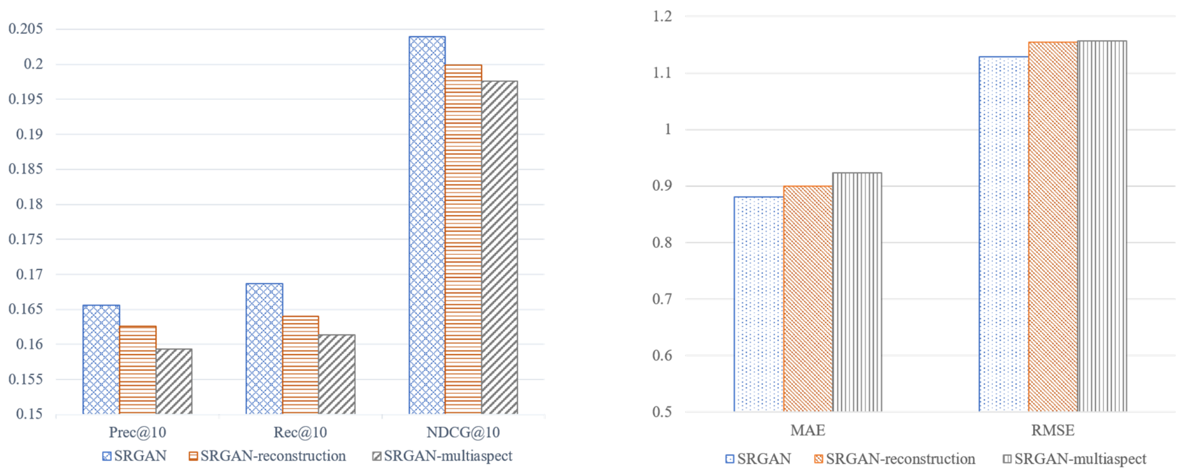
needs to be explored through the ablation experiment.

Table 6. The experimental results of recommendation model based on generative adversarial network.

Datasets	Metric	GCGAN	CFGAN	RSGAN	ESRF	SRGAN
	Precision@10	0.1233	0.1485	0.1535	<u>0.1636</u>	**0.1656**
	Recall@10	0.1369	0.1538	0.1562	<u>0.1653</u>	**0.1686**
Last.FM	NDCG@10	0.1494	0.1865	0.1910	<u>0.2004</u>	**0.2039**
	MAE	0.8904	0.9987	0.9291	<u>0.8960</u>	**0.8803**
	RMSE	1.1457	1.3411	1.2516	<u>1.1531</u>	**1.1284**
	Precision@10	0.1207	0.1576	0.1726	<u>0.1823</u>	**0.1847**
	Recall@10	0.0402	0.0513	0.0603	<u>0.0654</u>	**0.0670**
Douban	NDCG@10	0.1524	0.1874	0.1926	<u>0.2103</u>	**0.2132**
	MAE	0.8979	1.2353	0.9571	<u>0.9001</u>	**0.8853**
	RMSE	1.2532	1.5490	1.3897	<u>1.2564</u>	**1.2407**

4.4. Ablation experiment

In order to verify the influence of each module on the model performance, two variant models, SRGAN-Reconstruction and SRGAN-Multiaspect, are proposed here. SRGAN-reconstruction is the variant model after removing the social reconstruction module, and SRGAN-Multiaspect is the variant model after removing the multi-angle influence layer. The ablation experiment results on Last.FM dataset and Douban dataset are shown in Figures 2 and 3, respectively.



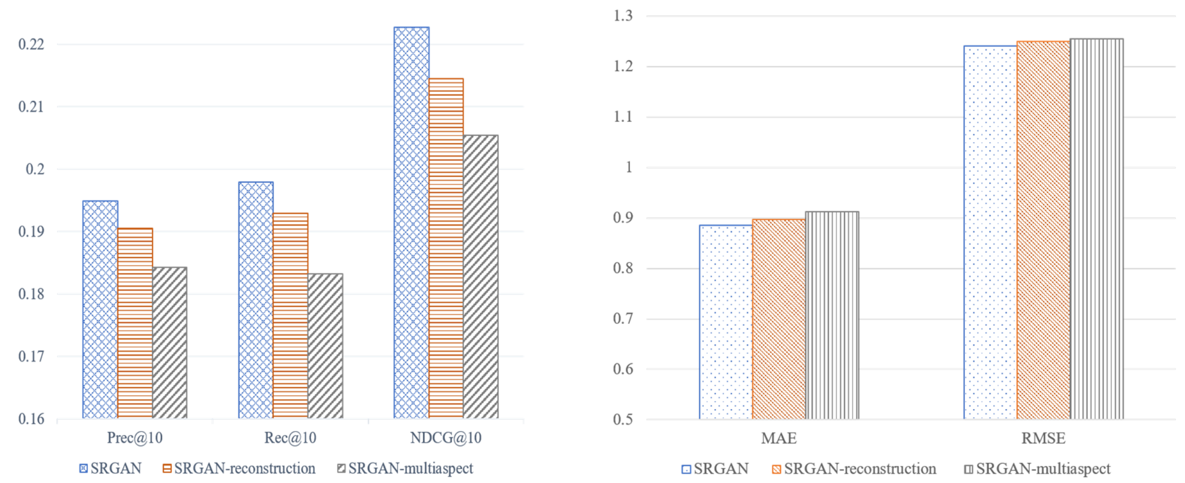
(a)The results of ranking metrics on Last.FM (b) The results of rating metrics on Last.FM

Figure 2. The results of ablation experiments on Last.FM dataset.

By analyzing the experimental results shown in Figures 2 and 3, the following can be observed:

After removing the social reconstruction module and multi-angle influence layer, the experimental results of each metric on Last.FM and Douban datasets become worse, indicating that the above two parts have a positive impact on the performance of the model. The social reconstruction module is introduced to dynamically adjust the weight of users' social relations with their surroundings and constantly optimize their social relations. When the social neighborhood is used to assist recommendation, more effective information can be obtained, so as to select the friends who are really similar to the users' preferences, making the recommendation results more and more accurate.

In addition, it can be found that the performance of the model is reduced when the multi-angle influence layer is removed, indicating that users should distinguish when adopting the opinions of friends. When users interact with different items, friends have different effects on users, and users should assign greater weight to friends who know about the current item.



(a)The results of ranking metrics on Douban (b) The results of rating metrics on Douban

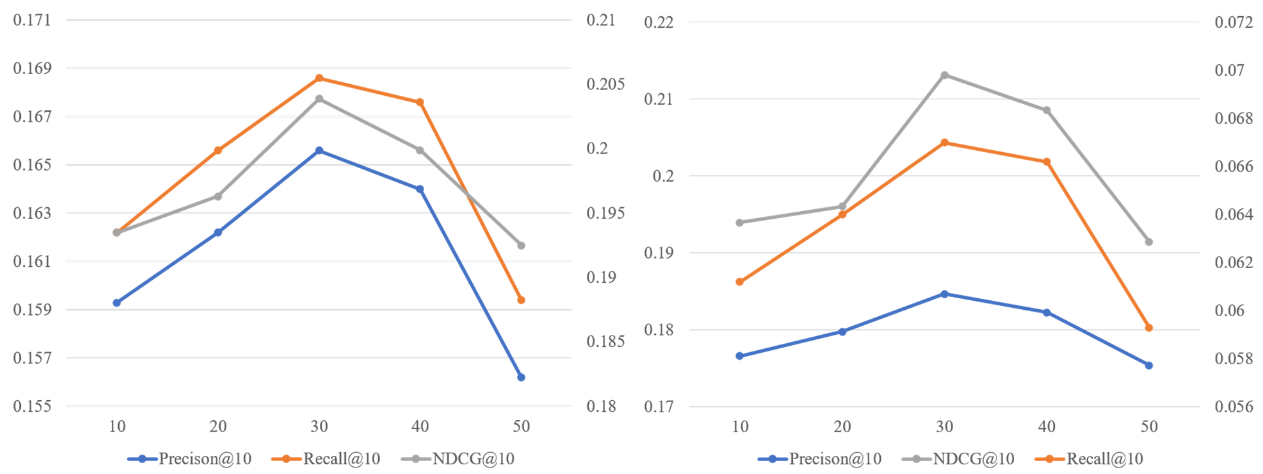
Figure 3. The results of ablation experiments on Douban dataset.

4.5. Parameter sensitivity analysis

4.5.1. The Influence of the number of friends selected

The friends filtering layer will select K friends into the multi-angle influence layer to integrate the influence of friends on users to assist recommendation, and the number of friends will directly affect the final experimental results of the model. Therefore, we selected different K values through experiments to explore their influence on the experimental results. The experimental results on Last.FM and Douban datasets are shown in Figure 4.

In order to show the results of Precision@10, Recall@10 and NDCG@10 changing with the number of friends clearly in the same graph, the x-coordinate is set as the number of selected friends, and the y-coordinate is the evaluation value. Here, the y-coordinate adopts primary and secondary axes. The blue line represents Precison@10, the orange line represents Recall@10, and the gray line represents NDCG@10. In Figure 4(a), the values of Precison@10 and Recall@10, are based on the left principal axis. NDCG@10 is the secondary axis to the right. In Figure 4(b), the values of Precision@10 and NDCG@10 are based on the left primary coordinate axis, and the values of Recall@10 are based on the right secondary coordinate axis.



(a) The influence of K values on Last.FM (b) The influence of K values on Douban

Figure 4. The influence of different values of K on the model performance.

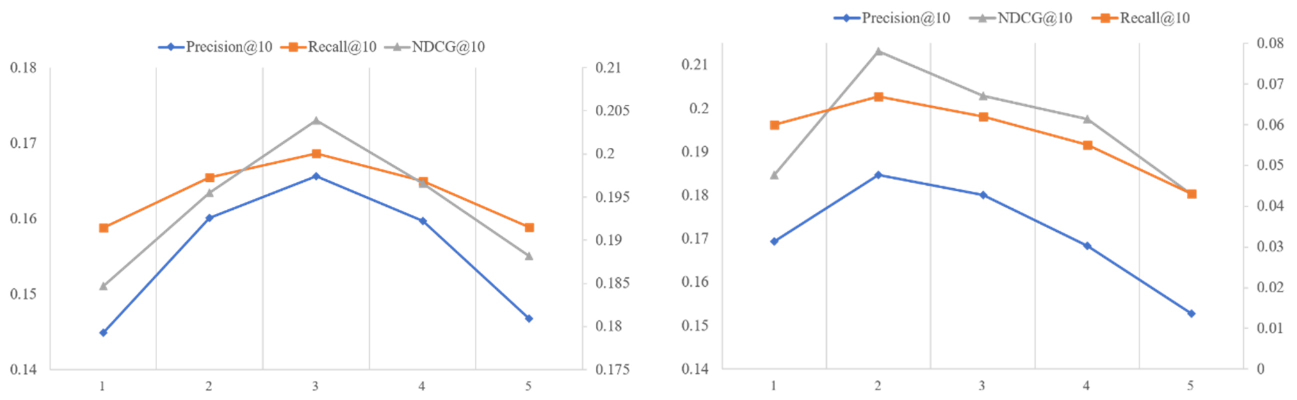
By analyzing the experimental results in Figure 4, it can be observed that the experimental results of the SRGAN model are affected by the number of friends, and it shows similar trends in the two datasets: At first, the values of the evaluation metrics increase significantly with the increase of the number of friends. When the number of friends is 30, the optimal results are obtained in both Last.FM and Douban datasets. When the number of friends is over 30, the results decrease slightly, and they decrease significantly when the number is 50. This indicates that when the number of selected friends is too small, we cannot get enough information to help recommend. When the number of selected friends is too large, it is easy to select friends that are significantly different from users' interests and preferences, thus introducing noise and reducing the performance of the model.

4.5.2. The influence of the number of aggregation layers

We use graph convolutional networks to aggregate information of higher-order neighbors in both social graph and user item interaction graph. As GCNs are sensitive to the number of aggregation layers, we investigated the influence of the number of layers L in this section. The experimental results on Last.FM and Douban datasets are shown in Figure 5.

In order to show the results of Precision@10, Recall@10 and NDCG@10 changing with the number of layers clearly in the same graph, the x-coordinate is set as the number of aggregation layers, and the y-coordinate is the evaluation value. Here, the y-coordinate adopts primary and secondary axes. The blue line represents Precison@10, the orange line represents Recall@10, and the gray line represents NDCG@10. In Figure (a), the values of Precison@10 and Recall@10, are based on the left principal axis. NDCG@10 is the secondary axis to the right. In Figure (b), the values of Precision@10 and NDCG@10 are based on the left primary coordinate axis, and the values of Recall@10 are based on the right secondary coordinate axis.

It can be seen that the experimental results on Last.FM and Douban datasets first rise and then decline with the increase of the number of aggregation layers. Among them, on the Last.FM dataset, when the number of aggregation layers is 3, the model results are optimal. On the Douban dataset, the optimal results have been obtained when the number of aggregation layers is 2. This indicates that more effective information cannot be obtained by aggregating only one order of neighbors. However, when the number of aggregation layers is too high, the aggregated higher-order neighbor information is inaccurate, thus introducing noise and reducing model performance.



(a) The influence of L values on Last.FM (b) The influence of L values on Douban

Figure 5. The influence of different values of *L* on the model performance.

5. Conclusions and future work

In this paper, we propose a recommendation model SRGAN based on generative adversarial network and social reconstruction. The generator generates friends' opinions similar to users' personal interests and preferences, and the discriminator distinguishes friends' opinions from users' personal interests. In the multi-angle social influence layer of the generator, the influence of friends on users is divided into the overall preference influence and the specific preference influence, and the different influences of friends on users when interacting with different items is considered. Generators and discriminators form a confrontation and promote each other in training. Through social reconstruction module, social networks are constantly reconstructed to optimize users' social networks. Finally, in order to verify the effectiveness of the proposed model, two sets of comparison experiments were conducted to compare the recommendation models using social information and generating adversarial networks. The experimental results show that the results of SRGAN on the four datasets are better than the baseline models, indicating that SRGAN effectively utilizes social relations and alleviates the problem of data sparsity. In addition, it can be found through the variant experiment results that the performance of the model decreases after removing the social reconstruction module and the multi-angle influence layer, indicating that these two parts have a positive impact on the experimental results. The social reconstruction module optimizes the social relations of users, making the opinions of friends more and more accurate. In addition, the multi-angle social influence layer will distinguish the influence of friends and assign greater weight to the friends who know the current item, thus improving the recommendation performance.

Acknowledgments

This work was supported by the Key Research and Development Projects in Hebei Province [20310802D] and the Innovation Capability Improvement Project in Hebei Province [22567603H].

Conflict of interest

The authors declare there is no conflict of interest.

References

1. Z. Batmaz, A. Yurekli, A. Bilge, A review on deep learning for recommender systems: challenges and remedies, *Artif. Intell. Rev.*, **52** (2019), 1–37. https://doi.org/10.1007/s10462-018-9654-y
2. K. Mariusz, W. Alicja, S. Krzysztof, S. Wlodzimierz, Beyond the Big Five personality traits for music recommendation systems, *EURASIP J. Audio Speech Music Process.*, **2023** (2023), 1–17. https://doi.org/10.1186/s13636-022-00269-0
3. Y. Liu, J. Miyazaki, Knowledge-aware attentional neural network for review-based movie recommendation with explanations, *Neural Comput. Appl.*, **35** (2022), 1–19. https://doi.org/10.1162/neco_a_01548
4. C. Ju, J. Wang, G. Zhou, The commodity recommendation method for online shopping based on data mining, *Multimedia Tools Appl.*, **78** (2019), 30097–30110. https://doi.org/10.1007/s11042-018-6980-7
5. B. Xu, H. Lin, L. Yang, Y. Lin, K. Xu, Cognitive knowledge-aware social recommendation via group-enhanced ranking model, *Cognit. Comput.*, **14** (2022), 1055–1067. https://doi.org/10.1007/s12559-022-10001-x
6. J. Liao, W. Zhou, F. Luo, J. Wen, X. Li, J. Zeng, SocialLGN: Light graph convolution network for social recommendation, *Inf. Sci.*, **589** (2022), 595–607. https://doi.org/10.1016/j.ins.2022.01.001
7. L. Wang, Y. Xiong, Y. Li, Y. Liu, A collaborative recommendation model based on enhanced graph convolutional neural network, *J. Comput. Res. Dev.*, **58** (2021), 1987–1996.
8. Y. Zeng, Q. Mou, L. Zhou, A recommendation model of session perception based on Graph presentation learning, *J. Comput. Res. Dev.*, **57** (2020), 590–603.
9. R. Ge, S. Chen, Graph convolutional network for recommender systems, *J. Software*, **31** (2020), 1101–1112. https://doi.org/10.13328/j.cnki.jos.005928
10. I. Goodfellow, J. Pouget-Abadie, M. Mehdi, B. Xu, D. Warde-Farley, S. Ozair, et al., Generative adversarial networks, *Commun. ACM*, **63**(2014), 139–144. https://doi.org/10.1145/3422622
11. S. Wang, H. Jin, J. Sun, GAN image adversarial sample generation method, *Comput. Sci. Explor.*, **15** (2021), 702–711.
12. G. Chen, Y. Liu, S. Zhong, X. Zhang, Musicality-novelty generative adversarial nets for algorithmic composition, in *Proceedings of the 26th ACM international conference on Multimedia*, **2018** (2018), 1607–1615. https://doi.org/10.1145/3240508.3240604
13. T. Bu, Z. Yang, S. Jiang, G. Zhang, H. Zhang, L. Wei, 3D conditional generative adversarial network-based synthetic medical image augmentation for lung nodule detection, *Imaging Syst. Technol.*, **31** (2021), 670–681. https://doi.org/10.1002/ima.22511

14. J. Li, J. Li, C. Wang, X. Zhao, Wide & deep generative adversarial networks for recommendation system, *Intell. Data Anal.*, **27** (2023), 121–136. https://doi.org/10.3233/IDA-216400
15. C. Zhang, Y. Wang, L. Zhu, J. Song, H. Yin, Multi-graph heterogeneous interaction fusion for social recommendation, *ACM Trans. Inf. Syst.*, **40** (2021), 1–26. https://doi.org/10.1145/3466641
16. M. Jamali, M. Ester, A matrix factorization technique with trust propagation for recommendation in social networks, in *Proceedings of the Fourth ACM Conference on Recommender Systems*, 2010. https://doi.org/10.1145/1864708.1864736
17. W. Fan, Y. Ma, Q. Li, Y. He, Y. Zhao, J. Tang, et al., Graph neural networks for social recommendation, in *The World Wide Web Conference*, **2019** (2019), 417–426. https://doi.org/10.1145/3308558.3313488
18. L. Wu, P. Sun, Y. Fu, R. Hong, X. Wang, M. Wang, A neural influence diffusion model for social recommendation, in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, **2019** (2019), 235–244.
19. W. Hamilton, R. Ying, J. Leskovec, Inductive representation learning on large graphs, *Adv. Neural Inf. Process. Syst.*, **2017** (2017), 30.
20. L. Wu, J. Li, P. Sun, R. Hong, Y. Ge, M. Wang, DiffNet++: A neural influence and interest diffusion network for social recommendation, *IEEE Trans. Knowl. Data Eng.*, **34** (2020), 4753–4766. https://doi.org/10.1109/TKDE.2020.3048414
21. L. Li, Z. Gan, Y. Cheng, J. Liu, Relation-aware graph attention network for visual question answering, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, **2019** (2019), 10313–10322.
22. Z. Yang, J. Qin, C. Lin, Y. Chen, R. Huang, Y. Qin, GANRec: A negative sampling model with generative adversarial network for recommendation, *Expert Syst. Appl.*, **214** (2023), 119–155. https://doi.org/10.1016/j.eswa.2022.119155
23. J. Wang, L. Yu, W. Zhang, Y. Gong, Y. Xu, B. Wang, et al., 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*, **2017** (2017), 515–524. https://doi.org/10.1145/3077136.3080786
24. D. Chae, J. Kang, S. Kim, J. 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*, **2018** (2018), 137–146. https://doi.org/10.1145/3269206.3271743
25. T. Sasagawa, S. Kawai, H. Nobuhara, Recommendation system based on generative adversarial network with graph convolutional layers, *J. Adv. Comput. Intell. Intell. Inf.*, **25** (2021), 389–396. https://doi.org/10.20965/jaciii.2021.p0389
26. J. Yu, M. Gao, H. Yin, J. Li, C. Gao, Q. Wang, Generating reliable friends via adversarial training to improve social recommendation, in *2019 IEEE International Conference on Data Mining (ICDM)*, **2019** (2019), 768–777. https://doi.org/10.1109/ICDM.2019.00087
27. J. Yu, H. Yin, J. Li, M. Gao, Z. Huang, L. Cui, Enhance social recommendation with adversarial graph convolutional networks, *IEEE Trans. Knowl. Data Eng.*, **34** (2020), 3727–3739. https://doi.org/10.1109/TKDE.2020.3033673

28. T. Zhao, J. McAuley, 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*, **2014** (2014), 261–270. https://doi.org/10.1145/2661829.2661998
29. H. Liu, J. Wen, L. Jing, J. Yu, Leveraging implicit social structures for recommendation via bayesian generative model, *Sci. China Inf. Sci.*, **65** (2022), 1–3. https://doi.org/10.1007/s11432-019-2884-0
30. Y. Yu, W. Qian, L. Zhang, R. Gao, A graph-neural-network-based social network recommendation algorithm using high-order neighbor information, *Sensors*, **22** (2022), 7122. https://doi.org/10.3390/s22197122
31. J. Lin, S. Chen, J. Wang, Graph neural networks with dynamic and static representations for social recommendation, in *Database Systems for Advanced Applications: 27th International Conference*, **13246** (2022), 264–271. https://doi.org/10.1007/978-3-031-00126-0_18