



TagRec: Trust-Aware Generative Adversarial Network with Recurrent Neural Network for Recommender Systems

Honglong Chen¹(✉), Shuai Wang¹, Nan Jiang², Zhe Li¹, Na Yan¹, and Leyi Shi³

¹ College of Control Science and Engineering, China University of Petroleum, Qingdao, China
chenhl@upc.edu.cn

² College of Information Engineering, East China Jiaotong University, Nanchang, China

³ College of Computer Science and Technology, China University of Petroleum, Qingdao, China

Abstract. Recently recommender systems become more and more significant in the daily life. Although the recommender systems based on the generative adversarial network (GAN) are competent, the user trust information is seldom taken into consideration. In this paper, we propose a Trust-Aware Generative adversarial network with recurrent neural network for RECommender systems named *TagRec*, which makes use of the user trust information for top-N recommendation. In the framework, the discriminative model is a multi-layer perceptron to distinguish whether a sample is from the real data or fake data. The discriminator helps to guide the training of the generative model to make it fit the data distribution of the user trust information. The generative model is a recurrent neural network (RNN) with long short-term memory (LSTM) cells, aiming to confuse the discriminative model by generating samples as similar as possible to the real data. Through the adversarial training between the discriminative and generative models, the user trust information can be fully used to improve the recommendation performance. We conduct extensive experiments on real-word datasets to validate the effectiveness of the TagRec.

Keywords: Recommender systems · Generative adversarial network · Recurrent neural network · Trust-aware

1 Introduction

The current explosive growth of data makes the information overload more and more serious, which can be solved by recommender systems. Matrix factorization (MF) [7] is one of the most classical recommendation methods, in which the recommendation is based on the interactive information between users and items. Although MF has been widely used in recommender systems due to its simplicity and attractive accuracy, it still suffers from two main problems. The first one is the severe data sparsity, and the other one lies in its failing to well learn the deep representation of interactive information between the users and items. The performance of recommender systems will be severely constrained by the above two problems. The user trust information can be

used to increase data volume based on the rating information for the data sparsity. Deep learning provides a potential solution for the deep representation of data, which can automatically learn the feature representation from heterogeneous data through multi-layer nonlinear network structure. In this paper, we intend to use the social information and deep learning in the recommender systems.

Recently, Goodfellow *et al.* [1] proposed Generative Adversarial Network (GAN), which learns to fit the distribution of given data by adversarial training. Inspired by GAN, IRGAN [3] is a unified framework that takes advantage of both generative model and discriminative model to apply the adversarial training in recommender systems. GraphGAN [4] also adopted the adversarial training and proposed a novel graph softmax to overcome the limitations of traditional softmax function. Both the IRGAN and GraphGAN use the policy gradient instead of gradient descent for the model optimization since the sampling of discriminative model is discrete. A novel GAN-based collaborative filtering framework called CFGAN [2] was proposed, which can solve the fundamental problem of the GAN-based methods, *i.e.*, the limitation of discrete item index generation. However, CFGAN ignores the potential of social information in recommender systems.

To address this problem, we propose a Trust-Aware Generative adversarial network with recurrent neural network for RECommender systems named *TagRec*, which combines the social information in the recommender systems. The proposed TagRec consists of two parts: a generative model G and a discriminative model D . In the generative model, the dynamic recurrent neural network with long short-term memory (LSTM) cells is adopted. A user rating trust sequence is firstly constructed based on the user rating vector according to the user trust relationship. Then based on the historical user rating information and user trust information, the rating trust sequence is input to G to generate a recommendation list for the user. In the discriminative model, the multi-layer perceptron is adopted. The real data and fake data (generated by the generator) are input to D , which aims to distinguish them. With the adversarial training between the discriminative and generative models, the discriminator helps to guide the training of the generative model to make it fit the data distribution of user trust information.

The main contributions of this paper are summarized as follows:

- By making use of the users' social relationship, we construct a user rating trust sequence based on the user rating vector, in which the similarity calculation is adopted to avoid the curse of dimensionality.
- We propose to integrate the generative adversarial network and recurrent neural network, which can extract both the rating information features and social information features, to handle the sparsity problem in the recommender systems.
- We implement the proposed TagRec model and conduct extensive experiments on two real-world datasets to validate its effectiveness.

The rest of this paper is organized as follows. In Sect. 2, we briefly review the related work. In Sect. 3, we introduce the proposed social recommendation method TagRec in details. In Sect. 4, we describe the experimental data, implementation details, evaluation metrics and benchmarks. In Sect. 5, we analyze the experimental results. In Sect. 6, we conclude this paper and put forward the future work.

2 Related Work

2.1 Recurrent Neural Network Based Recommendation

The Recurrent Neural Network (RNN) can be used to process the sequential data of the recommender systems. In [13], RNN is applied to the session-based recommender systems, which adopts Gated Recurrent Unit (GRU) as the RNN unit and takes the first clicked item as the initial input of the GRU unit. Each click of the user will produce a recommendation result that depends on all the previous clicks. A dynamic model is proposed in [6], which is combined with RNN to predict the future behavioral trajectories of users. This work adopts the long short-term memory (LSTM) as the basic RNN unit and uses LSTM to learn the states of the users and items respectively. In [15], Liu *et al.* propose a context-aware recurrent neural network to address the problem of context-aware sequential recommendation. In the proposed TagRec, we use RNN combined with GAN to process the sequential data in recommender systems.

2.2 Generative Adversarial Network Based Recommendation

Recently, GAN has been successfully applied to the recommendation tasks. IRGAN [3] is a unified framework, which takes advantage of both the generative model and discriminative model and can be used in the web search, item recommendation and question answering. In [16], RecGAN is proposed, which combines RNN and GAN to improve recommendation performance. Different from the proposed TagRec in this paper, RecGAN leverages RNN to extract time feature from the interactive information between the users and items, while TagRec is to process the social information. In [2], a new vector-wise mechanism is proposed to improve the training of recommendation methods based on policy gradient without considering the social information.

3 The Proposed Framework

In this paper, we focus on the top-N recommendation problem in recommender systems. Let $\mathcal{U} = \{u_1, u_2, \dots, u_m\}$ represent the set of users and $\mathcal{I} = \{i_1, i_2, \dots, i_n\}$ denote the set of items, where m is the number of users and n is the number of items respectively. Let $R = [R_{u,i}]_{m \times n}$ denote the ratings expressed by the users on items where $R_{u,i}$ is a real number that represents the preference of user u on item i . The larger the value of $R_{u,i}$ is, the more user u likes item i . The value of $R_{u,i}$ ranges from 1 to 5. We set $R_{u,i}$ to 1 if $R_{u,i} > 1$, and 0 otherwise. In addition to the rating matrix R , each user has a social trust matrix $T = [T_{u,v}]_{m \times m}$ where $T_{u,v}$ denotes the trust value that the trustor u has on trustee v . Usually the trust value is either 0 or 1, where 0 means that user u has no trust with user v and 1 means that user u completely trusts user v . The task of the recommender system is to use the social trust matrix T and the existing values in the rating matrix R to predict the missing values in R .

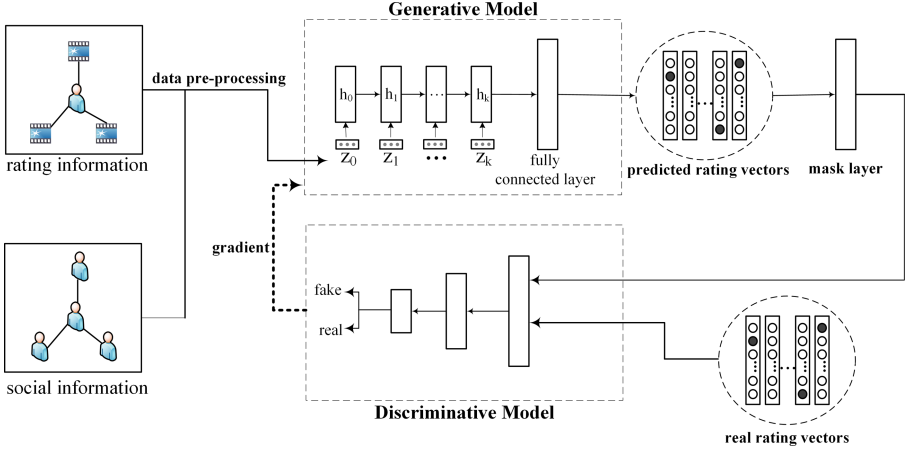


Fig. 1. The overall framework of the proposed TagRec.

3.1 An Overview of the Proposed Framework

The architecture of the proposed framework is shown in Fig. 1, which consists of three parts: data pre-processing, adversarial learning and negative sampling. Data pre-processing constructs a user rating trust sequence as the input of the proposed framework TagRec based on the user rating vector and social relationship. Adversarial learning consists of generative model and discriminative model. The generative model generates fake data as the negative sample of the discriminative model. The purpose of the discriminative model is to distinguish whether a sample is from the real data or generated data. Negative sampling is to solve the output polarizing problem of the generative model by adding a mask layer and a negative sampling function.

3.2 Data Pre-processing

In order to merge the user rating matrix R and user trust matrix T into one matrix without losing the deep representation between them, we exploit the user set \mathcal{U} . In fact, the user set \mathcal{U} bridges the gap between the matrix R and matrix T . Each user u in the user set \mathcal{U} is also in matrix R and matrix T . This gave us the inspiration to merge the data. Each user u has a rating vector of n dimensions $[R_u]_n$ in matrix R for all items. Each user u has a trust vector of m dimensions $[T_v]_m$ for all users. Therefore, for each element in the trust vector $[T_v]_m$ we can replace it with the n dimensions rating vector corresponding to the user u_m . By this means, we can obtain a new matrix of $m \times m \times n$ dimensions for m users in matrix T . We define this new matrix as the trustee rating matrix $TR = [TR_{u,v,i}]_{m,m,n}$. The trustee rating matrix successfully embeds trust relationship into user rating by deep representation. However, due to the following two factors, too much consideration of other trustees may even reduce the performance of recommender systems. First, it is computationally expensive. Second, even if a trustor has a lot of trustees, his/her preference will only be affected by some of the most trusted

users [14]. To solve the above problems, we calculate the similarity between a given user and his/her trustees, and then only select the trustees with high similarity as the input data in matrix TR . We use Jaccard index [17] $J(u, u_m)$ to calculate the similarity between user u and trustees u_m :

$$J(u, u_m) = \frac{|R_u \cap R_{u_m}|}{|R_u \cup R_{u_m}|}. \quad (1)$$

The number of the selected trustees is defined as k , and the effect of k will be discussed in Sect. 5. Therefore, the dimension of matrix TR will be $m \times k \times n$. In the proposed framework TagRec, matrix TR is the input and the output is the prediction rating matrix for users. In this way, the framework can learn the deep representation between user rating and trust relationship. The whole process simulates the consensus that users' preference can be inferred from their trustees.

3.3 Adversarial Learning for Recommendation

The generative adversarial network consists of a generator G and a discriminator D . Let t_u and r_u denote user u 's trust information and existing rating information respectively. Let r_i and \tilde{r}_i denote the true and predicted ratings on item i respectively. θ_G and ϕ_D are denoted as the model parameters of the generator G and the discriminator D respectively. The proposed framework is to learn the following two models:

Generator $G(r_i|t_u, r_u; \theta_G)$, which tries to approximate the real rating data distribution over items $p_{true}(r_i|t_u, r_u)$, and generate the most similar data distribution $p_\theta(r_i|t_u, r_u)$.

Discriminator $D(r_i, \tilde{r}_i; \phi_D)$, which attempts to distinguish the real data distribution $p_{true}(r_i|t_u, r_u)$ from the fake data distribution $p_\theta(r_i|t_u, r_u)$ generated by G .

$$\begin{aligned} \min_{\theta_G} \max_{\phi_D} V(G, D) = & \mathbb{E}_{r_i \sim p_{true}(r_i|t_u, r_u)} [\log D(r_i|t_u, r_u; \phi_D)] \\ & + \mathbb{E}_{\tilde{r}_i \sim p_\theta(r_i|t_u, r_u)} [\log (1 - D(\tilde{r}_i|t_u, r_u; \phi_D))]. \end{aligned} \quad (2)$$

Based on Eq. (2), the generator and the discriminator learn the optimal model parameters by iterative training.

Generative Model. The framework of TagRec adopts recurrent neural network (RNN) as the generative model. As mentioned in Sect. 3.2, a user's predicted ratings on items can be inferred from his or her trustees' ratings. Since the influence of trustees on users is cumulative, the function of recurrent neural network is similar with this situation. The RNN not only considers the input of the current moment, but also gives the network a 'memory' of the previous content. By taking advantage of this feature, RNN can capture the deep representation of the influence of each trustee on the user. As the user does not declare the trust relationship to all other users, the second dimension of the trustee rating matrix TR is uncertain in practice. Therefore, we adopt the structure of dynamic RNN and use long short-term memory (LSTM) cells as the basic unit. In the data pre-processing stage, the trust information and rating information can be fused

into a trustee rating matrix $TR = [TR_{u,v,i}]_{m,k,n}$. The TagRec takes the n -dimensional vectors $[TR_i]_n$ in the matrix TR as the input of dynamic RNN at time t , where the maximum value of t is equal to the number of the selected trustees k . LSTM unit consists of an input gate i_t , a forget gate f_t , an output gate o_t , and a state cell c_t . These gates can be calculated based on the previous hidden state h_{t-1} and the current input z_t :

$$f_t, i_t, o_t = \sigma(W[h_{t-1}, z_t]) \quad (3)$$

where $\sigma(\cdot)$ is the sigmoid activation function, W is the weight parameter. A tanh layer is used to output the updated content \tilde{c}_t and the cell state c_{t-1} at the previous moment $t - 1$ is updated to the cell state c_t at time t :

$$\tilde{c}_t = \tanh(W_c[h_{t-1}, z_t]) \quad (4)$$

$$c_t = f_t \times c_{t-1} + i_t \times \tilde{c}_t \quad (5)$$

where $\tanh(\cdot)$ is the hyperbolic tangent activation function, W_c is the weight of the tanh layer. The hidden state at time t can be given by:

$$h_t = o_t \times \tanh(c_t). \quad (6)$$

In order to reconstruct the rating matrix from the deep representation, a full connected layer is defined as:

$$f_s = W_s h_t + b_s, \quad (7)$$

where W_s and b_s are the weight and bias of the full connected layer respectively and f_s is the predicted rating matrix generated by the generator. For learning the optimal parameters θ^* of the generator, the framework needs to fix the discriminator's parameters and the generator can be optimized by minimizing the following function:

$$\theta^* = \arg \min_{\theta_G} \sum_{u \in \mathcal{U}} \mathbb{E}_{\tilde{r}_i \sim p_{\theta}(r_i | t_u, r_u)} [\log(1 - D(\tilde{r}_i | r_u, t_u; \phi_D))]. \quad (8)$$

Since the data generated by the generator in the TagRec is continuous vector in range $[0, 1]$, the gradient descent can be used for training directly.

Discriminative Model. The proposed framework adopts multi-layer perceptron as the discriminative model. The purpose of discriminative model is to distinguish the real data from the fake data generated by the generative model. This is actually a simple binary classification problem. Therefore, the discriminative model does not need too complex network, but only needs the non-linear learning ability. Multi-layer perceptron can learn the deep representation of the data well by changing the number of layers and neurons in the neural network [12], and it is easy to train with gradient descent. The discriminator takes the real user rating vector and the predicted rating vector generated by the generator as inputs. It should be noted that different from CFGAN, the real user rating vector defined in TagRec is not the original user rating vector, but the

processed vector based on the original user rating vector plus the rating vector of the user’s trustees. The function of discriminator can be summarized as follows:

$$a_l = \sigma \left(W_l a_{l-1} + b_l \right), \quad (9)$$

where W_l denotes the l -th layer’s weight and b_l denotes the l -th layer’s bias. a_l denotes the output of the l -th layer and a_{l-1} is the output of the previous layer. In particular, when $l = 1$, $a_{l-1} = a_0$ is the original input data of the discriminator. For learning the optimal parameters, the framework need to fix the generator’s parameters and the discriminator can be optimized by maximizing the following function:

$$\begin{aligned} \phi^* = \arg \max_{\phi_D} \sum_{u \in \mathcal{U}} & \left(\mathbb{E}_{r_i \sim p_{tr u e}(r_i | t_u, r_u)} [\log D(r_i | t_u, r_u; \phi_D)] \right. \\ & \left. + \mathbb{E}_{\tilde{r}_i \sim p_{\theta}(r_i | t_u, r_u)} [\log(1 - D(\tilde{r}_i | t_u, r_u; \phi_D))] \right), \quad (10) \end{aligned}$$

where ϕ^* denotes the optimal parameters of the discriminator. The discriminator can be trained with gradient descent.

3.4 Negative Sampling

Through the data pre-process and adversarial learning illuminated above, the framework of TagRec can learn the deep representation between the rating and social information. But as described in CFGAN [2], the vector-wise method will face an output polarizing problem that the generator may simply predict all the outputs in the predicted rating matrix as 1 due to the lack of negative samples. Therefore, negative sampling is adopted and next we will introduce the solution in details.

As shown in the Fig. 1, similar to CFGAN, a mask layer is added on the top of the generator. The layer is defined as a matrix e_u , which has the same dimensions $m \times n$ as the output matrix of the generator. The mask layer matrix e_u needs to be generated in two steps. Firstly, a matrix is defined as e_u'' which has $m \times n$ dimensions and all the entries in the matrix are 0. Each element $e_{m,n}''$ in the matrix corresponds to the rating that the user u_m rated on the item i_n . For the elements corresponding to those items that the user and user’s trustees have evaluated, the value is changed to 1 if it is greater than 1 and the new matrix is defined as e_u' . The output of the generator is multiplied by the corresponding elements of the matrix e_u' . By this step, the framework will only train the existing rating data and ignore the items that the users and corresponding trustees have not evaluated, which is similar to the principle of matrix factorization. Secondly, we select partial data from those items that the users and corresponding trustees have not evaluated as negative samples of the generator. The negative sampling ratio is defined as s . For the data sampled from unrated items, the corresponding element in the matrix e_u' is changed from 0 to 1 and the final new mask matrix is defined as e_u . After the second step, the framework can not only train the existing data, but also propagate the gradient back to the negative samples during the training process. Since unrated items’ rating is stored as 0 in the matrix TR , the sampled data is actually equivalent to negative samples of the generator during training. Finally, The output of the generator is multiplied by

the corresponding elements of the matrix e_u . In this way, the generator can produce low values on the negative items and the reconstructed loss function of the generator is denoted as:

$$\theta^* = \arg \min_{\theta_G} \sum_{u \in \mathcal{U}} \left(\mathbb{E}_{\tilde{r}_i \sim p_{\theta}(r_i|t_u, r_u)} \left[\log(1 - D(\tilde{r}_i \odot e_u | t_u, r_u; \phi_D)) \right] + \alpha \cdot \sum_j (x_{uj} - \tilde{x}_{uj})^2 \right), \quad (11)$$

where $\sum_j (x_{uj} - \tilde{x}_{uj})^2$ is the regularization term, α is the regularization coefficient and \odot presents the element-wise product. The reconstructed loss function of the discriminator is defined as follows:

$$\phi^* = \arg \max_{\phi_D} \sum_{u \in \mathcal{U}} \left(\mathbb{E}_{r_i \sim p_{true}(r_i|t_u, r_u)} [\log D(r_i | t_u, r_u; \phi_D)] + \mathbb{E}_{\tilde{r}_i \sim p_{\theta}(r_i|t_u, r_u)} [\log(1 - D(\tilde{r}_i \odot e_u | t_u, r_u; \phi_D))] \right). \quad (12)$$

During the training stage, the discriminator and the generator are trained alternatively in an adversarial manner via Eq. (12) and Eq. (11), respectively. The pseudo code of the proposed TagRec is presented in Algorithm 1 as follows:

Algorithm 1. The procedure of TagRec

Input: Trustee rating matrix TR , learning rate for generator G and discriminator D , the real user rating vector, minibatch size for training.

Output: The predicted users' rating on items.

- 1: Data pre-processing.
 - 2: Calculate similarity and select k trustees.
 - 3: Initialize model parameters G_{θ} and D_{ϕ} .
 - 4: **repeat**
 - 5: **for** Each $u \in \mathcal{U}$ **do**
 - 6: Negative sampling.
 - 7: **end for**
 - 8: **for** G-steps **do**
 - 9: Sample minibatch of users.
 - 10: Generate predicted rating vector \tilde{r}_i .
 - 11: Update G_{θ} by gradient descent.
 - 12: **end for**
 - 13: **for** D-steps **do**
 - 14: Use current G_{θ} to generate negative samples.
 - 15: Get positive samples sampled from true data.
 - 16: Update discriminator D_{ϕ} by gradient descent.
 - 17: **end for**
 - 18: **until** Convergence.
-

4 Experimental Setup

4.1 Datasets

Table 1. Characteristics of the datasets.

Datasets	FilmTrust	Ciao
Users	1,508	7,357
Items	2,071	99,746
Ratings	35,497	278,483
Density of ratings	1.14%	0.0379%
Social relations	1,853	111,781
Sparsity of social relations	0.0814%	0.2065%

We conduct the experiments based on two real-world datasets: FilmTrust [10] and Ciao [11]. Both datasets include user rating information of items and social information between users. The detailed statistics are summarized in Table 1. For each dataset, we randomly split the whole data into a training set (80%) and a testing set (20%).

4.2 Implementation Details and Evaluation Metrics

For the generator with dynamic RNN, we adopt a single-layer LSTM with 500 hidden neurons. We set the maximum number of iterations to 200. For both the generator and discriminator, we use stochastic gradient descent with learning rate 1×10^{-4} to optimize the parameters. The number of the selected users k is set to 5 and the negative sampling ratio s is set to 0.001. In training process, we conduct mini-batch training with batch size 128. We employ two widely-used evaluation metrics to evaluate the top- N recommendation performance of the proposed TagRec, including Normalised Discounted Cumulative Gain ($NDCG@N$), Mean Average Precision ($MAP@N$). N is set to 5, 10 and 20.

4.3 Comparison to Baselines

We compare the performance of the proposed TagRec with the following benchmarks.

BPR [8]: It provides personalized recommendations by optimizing the ordering between two items based on Bayesian analysis. It is a baseline for top- N recommendation.

SBPR [9]: It is an improved algorithm based on BPR, which uses social connections to improve accuracy of recommender systems.

TBPR [5]: It is also an improved BPR. The difference between TBPR and SBPR is that the former distinguishes the different effects of strong and weak ties on the recommender systems.

IRGAN [3]: It is the first work combining with generative adversarial networks for recommender systems.

CFGAN [2]: It is a vector-wise method, which points out the problem existing in the adversarial learning recommender system based on policy gradient.

5 Experimental Results and Analysis

5.1 Experimental Results

Table 2. Performance comparison of different recommender systems.

Datasets	Metrics	Algorithms					
		BPR	SBPR	TBPR	IRGAN	CFGAN	TagRec
Filmtrust	NDCG@5	0.3711	0.4135	0.4179	0.4571	0.4619	0.4718
	NDCG@10	0.4232	0.4574	0.4673	0.5093	0.5120	0.5251
	NDCG@20	0.4974	0.5039	0.5120	0.5307	0.5688	0.5824
	MAP@5	0.3044	0.3397	0.3450	0.3622	0.3792	0.4179
	MAP@10	0.3183	0.3586	0.3655	0.4038	0.4062	0.4282
	MAP@20	0.3645	0.3871	0.4033	0.4403	0.4529	0.4755
Ciaos	NDCG@5	0.0124	0.0131	0.0136	0.0145	0.0152	0.0158
	NDCG@10	0.0139	0.0143	0.0152	0.0169	0.0189	0.0197
	NDCG@20	0.0187	0.0211	0.0208	0.0223	0.0232	0.0241
	MAP@5	0.0087	0.0096	0.0103	0.0115	0.0120	0.0127
	MAP@10	0.0101	0.0110	0.0108	0.0121	0.0125	0.0132
	MAP@20	0.0112	0.0123	0.0118	0.0127	0.0135	0.0142

The performance of all recommendation algorithms on two real-world datasets is shown in Table 2. From the experimental results, we can draw the following findings:

- (1) TagRec outperforms BPR. This indicates that TagRec can learn data distribution better because the proposed TagRec based on deep learning can fit non-linear data distribution better.
- (2) SBPR and TBPR outperform BPR. That is because BPR only uses the users' ratings on items, while SBPR and TBPR utilize both the users' social relations and ratings on items. At the same time, TagRec obtains better performance than SBPR and TBPR. This indicates that the TagRec with adversarial training is promising on the task of recommender systems.
- (3) IRGAN and CFGAN can achieve better performance than SBPR and TBPR due to their excellent ability in learning representations. At the same time, the results show that CFGAN outperforms IRGAN. Both CFGAN and IRGAN are based on generative adversarial network. The difference lies in that IRGAN utilizes policy gradient

for optimization because it generates discrete data while CFGAN generates continuous values and is optimized by stochastic gradient descent. This indicates that discrete items sampling proposed in IRGAN can be further improved in generative adversarial network because the original GAN is designed for differentiable values and CFGAN proposes an appropriate method to solve this problem.

- (4) The TagRec outperforms CFGAN in MAP and $NDCG$. Compared with CFGAN, TagRec incorporates social relations. The results show that social relation is significant for recommender systems and the TagRec improves the performance of the recommender system.

5.2 Social Relations Analysis

In this section, we investigate the impact of the number of the selected trustees on our proposed approach. We take the FilmTrust dataset and $MAP@20$ as examples. Since the user has a maximum of 60 trustees in FilmTrust dataset, we set k to 1, 3, 5, 10, 20, 30, 40, 50 and 60. As shown in Fig. 2, when we set k to 5, the value of $MAP@20$ reaches the maximum value, and as the value of k increases gradually, the value of $MAP@20$ decreases correspondingly at the same time. Although when k is greater than 30, the value of $MAP@20$ is no longer reduced quickly due to data sparsity, we can find that the time consumption is still increasing. Therefore, selecting the most similar trustees by similarity calculation for a given user is necessary to keep a balance between recommendation performance and time consumption.

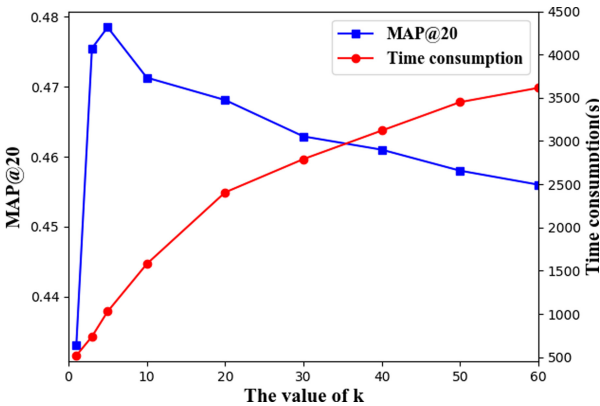


Fig. 2. Social relations analysis on FilmTrust dataset.

6 Conclusion and Future Work

In this paper, we propose a new recommendation method of combining social relationships based on generative adversarial networks. The deep representation between user-item interaction and social relations is learned by dynamic recurrent neural network with long short-term memory cells. The rating information and social relations

are integrated into an input matrix and user similarity is calculated to further improve performance. The experiments on two real-world datasets demonstrate the effectiveness of our method.

In the future, we will focus on extracting reliable users' social relationships more effectively. Although users claim some trustees for himself, this data volume is small and does not necessarily have a positive effect on the recommendation systems. Therefore we plan to study the issue of trustees' confidence in social network in the future.

Acknowledgments. This work was supported in part by NSFC grant (No.61772551), the Fundamental Research Funds for the Central Universities (No.18CX07003A), Natural Science Foundation of Shandong Province, China (No.ZR2019MF034), Research Fund for the Creative Research Team of Young Scholars at Universities in Shandong Province, grant No.2019KJN019, the International Exchange Project of China University of Petroleum (No.UPCIEF202052).

References

1. Goodfellow, I., et al.: Generative adversarial nets. In: *Advances in Neural Information Processing Systems*, Montreal, Canada, pp. 2672–2680 (2014)
2. Chae, D.K., Kang, J.S., Kim, S.W., Lee, J.T.: CFGAN: a generic collaborative filtering framework based on generative adversarial networks. In: *27th ACM International Conference on Information and Knowledge Management*, pp. 137–146. ACM, Torino (2018)
3. Wang, J., et al.: Irgan: a minimax game for unifying generative and discriminative information retrieval models. In: *40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 515–524. ACM, Tokyo (2017)
4. Wang, H.W., et al.: Graphgan: graph representation learning with generative adversarial nets. In: *32nd AAAI Conference on Artificial Intelligence*, New Orleans, USA (2018)
5. Wang, X., Lu, W., Ester, M., Wang, C., Chen, C.: Social recommendation with strong and weak ties. In: *25th ACM International on Conference on Information and Knowledge Management*, pp. 5–14. ACM, Indianapolis (2016)
6. Wu, C.Y., Ahmed, A., Beutel, A., Smola, A.J., Jing, H.: Recurrent recommender networks. In: *10th ACM International Conference on Web Search and Data Mining*, pp. 495–503. ACM, Cambridge (2017)
7. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. *Computer* **42**(8), 30–37 (2009)
8. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme L.: BPR: Bayesian personalized ranking from implicit feedback. In: *25th Conference on Uncertainty in Artificial Intelligence*, Montreal, Canada, pp. 452–461 (2009)
9. Zhao, T., McAuley, J., King, I.: Leveraging social connections to improve personalized ranking for collaborative filtering. In: *23rd ACM International Conference on Information and Knowledge Management*, pp. 261–270. ACM, Shanghai (2014)
10. Guo, G., Zhang, J., Yorke-Smith, N.: A novel bayesian similarity measure for recommender systems. In: *23rd International Joint Conference on Artificial Intelligence*, Beijing, China, pp. 2619–2625 (2013)
11. Guo, G., Zhang, J., Thalmann, D., Yorke-Smith, N.: ETAF: an extended trust antecedents framework for trust prediction. In: *International Conference on Advances in Social Networks Analysis and Mining*, Beijing, China, pp. 540–547 (2014)
12. Li, P.J., Wang, Z.H., Ren, Z.C., Bing, L.D., Lam, W.: Neural rating regression with abstract tips generation for recommendation. In: *40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 345–354. ACM, Tokyo (2017)

13. Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. In: 4th International Conference on Learning Representations, San Juan, Puerto Rico (2016)
14. Liu, C.Y., Zhou, C., Wu, J., Hu, Y., Guo, L.: Social recommendation with an essential preference space. In: 32nd AAAI Conference on Artificial Intelligence, New Orleans, USA (2018)
15. Liu, Q., Wu, S., Wang, D.Y., Li, Z.K., Wang, L.: Context-aware sequential recommendation. In: 16th International Conference on Data Mining, Barcelona, Spain, pp. 1053–1058 (2016)
16. Bharadhwaj, H., Park, H., Lim, B.Y.: Recgan: recurrent generative adversarial networks for recommendation systems. In: 12th ACM Conference on Recommender Systems, pp. 372–376. ACM, Vancouver (2018)
17. Jaccard, P.: The distribution of the flora in the alpine zone 1. *New Phytologist* **11**(2), 37–50 (1912)