SIGA: Social Influence Modeling Integrating Graph Autoencoder for Rating Prediction

Jinxin Liu[1,2], Yingyuan Xiao[1,2*], Wenguang Zheng[1,2*] and Ching-Hsien Hsu[3]

[1*]Engineering Research Center of Learning-Based Intelligent System, Ministry of Education, Tianjin University of Technology, Tianjin, 300384, China.
[2]Tianjin Key Laboratory of Intelligence Computing and Novel Software Technology, Tianjin University of Technology, Tianjin, 300384, China.
[3]College of Information and Electrical Engineering, Asia University, Taichung, 41354, Taiwan.

*Corresponding author(s). E-mail(s): yyxiao@tjut.edu.cn; wenguangz@tjut.edu.cn;
Contributing authors: jinxinliu@aliyun.com; chh@cs.ccu.edu.tw;

Abstract

With the revival of social networks, many studies try to integrate social relations of users to improve the accuracy of rating prediction. However, most existing methods cannot accurately reflect how social relations affect user preferences. The main reason is that these methods only investigate the directly or indirectly reachable nodes in social networks, while ignoring global transitivity of influence. Actually, for a particular user, his preference will not only affect his neighbors but also some people he does not know, especially when the person is an opinion leader. In this paper, we propose a social influence model, SIGA, which integrates graph autoencoder and is used for the rating prediction task. First, we establish a new method to quantify the social influence of users from the perspective of information dissemination in social networks. Second, we employ graph autoencoder (GAE) to model the interaction between users and items from the view of message passing on bipartite graph, which can obtain high-quality representations of users and items. By building a hybrid architecture of social modeling and

GAE, it is expected to be endowed with both benefits from them. In addition, our model is interpretable at both the structural level and attribute level. Extensive experimental results on four real-world datasets have shown the effectiveness and generalization of the proposed model.

Keywords: Social Influence, Recommendation System, Rating Prediction, Graph Autoencoder

1 Introduction

Recommendation systems (RS) are famous for their effectiveness in solving the problem of information overload and have become increasingly important in the era of information explosion. Rating prediction is one of the most significant tasks in RS, which directly affects the accuracy of the recommended results. The major objective of rating prediction is to explore users' preferences on different items and suggest a rating for each particular item. In order to improve the prediction accuracy, many studies try to mine the relations between users in a social network Fan et al (2019) Liu et al (2020). In the real world, when a user confuses which items to buy, he always asks his friends for suggestions. Thus, many researchers have claimed that the social influence of users is a key factor that should be investigated and integrated into rating prediction models. However, in order to make a better social recommendation system, there are some issues to be considered.

Global transitivity of influence. Generally, a social network can be formalized as a graph, where each node is a user, and each edge denotes the relation between two users. The number of edges from one node to another is called the connection distance between two nodes. At present, most existing social influence methods only consider the short-range connection in the graph, which means a user only affects his neighbors (e.g. friends), those who have a close connection distance with him in the graph. Actually, the influence of a user will transmit to his friends, friends of friends, even other strangers. Moreover, the opinions of some key opinion leaders (KOLs) will be widely spread, and even if you don't pay attention to them, you may see them in various advertisements and news. Ignoring global transitivity of influence, the model of social network is over-simplified, and cannot achieve high effectiveness.

Efficiency of quantifying influence. In the real world, authoritative people always enjoy higher influences and may affect more users' decisions or choices. Such a scenario should be considered in the modeling of the social network. The influence of each node in the graph of a social network should be distinguished and quantified. Besides, for a particular user, his influence on different users also should be specified. For example, a user generally has different influences on his friends, friends of friends, and other ones. On the other hand, when the number of nodes increases, the global quantification of influence is always accompanied by higher computational cost. Therefore, the traditional way of transferring influence by chain jumping in the neighborhood of users is not suitable for the global situation, and a new efficient social modeling method is needed.

In view of the above situation, in this paper, we investigate the rating prediction problem and propose a novel model, named SIGA, which integrates social influence modeling with graph autoencoder. The major contributions of our work are described as follows:

- We put forward a new social recommendation framework SIGA, which includes efficient global influence modeling, and we combine it with graph autoencoder to make rating prediction together.
- To the best of our knowledge, we are the first to model the influence of users in a social network by considering both global transitivity and quantification of users' influences.
- The graph autoencoder with GCN as encoder can build high-quality latent representation and benefit from the special structural features of the autoencoder, so the whole model has high generalization.
- Our proposed model is interpretable, and we also provide a proof.

The rest of this paper is organized as follows. The second section introduces some definitions defined in this paper. The third section describes our social influence modeling. The fourth section shows the details of SIGA. The fifth section is the experimental results. The sixth section summarizes the related work and the seventh section concludes this paper.

2 Preliminaries

In this section, we will introduce some notations and concepts used throughout the paper.

Link Prediction on Bipartite Graph. A bipartite graph is a kind of graph $\mathcal{G} = (\mathcal{W}, \ \mathcal{E})$, where $\mathcal{W}$ is a set of vertices, and $\mathcal{E}$ is a set of edges, and $\mathcal{W}$ can be divided into two disjoint parts $\mathcal{U}, \ \mathcal{V}$. For any edge in $\mathcal{E}$, one vertex belongs to $\mathcal{U}$ and the other belongs to $\mathcal{V}$, as shown in Fig. 1. Usually, there are a lot of gaps in the links on bipartite graphs, and lots of link prediction method is used to predict which node pairs will establish a new connection.

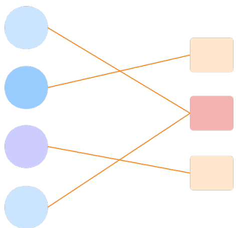


Fig. 1 Bipartite Graph. Circles on the left represent users, and squares on the right represent the items.

With the above definition, we now define the studied task.

Completion of the Rating Matrix. Given a data set $\mathcal{D}$, which contains the user's rating of items, we need to complete all the vacant parts. That is, for a given user u_i, we want to infer his most likely rating for each item v_j. Note that it is straight forward to convert the above task into selecting the optimal rating r_j^* with the highest conditional probability:

$$r_j^* = argmax_{r_j} \ \Pr \ (r_j \mid u_i, \ \mathcal{D} \ ) \tag{1}$$

The rating matrix completion task is formulated as a link prediction problem on a bipartite graph, each type of edge in the bipartite graph corresponds to a type of rating in the matrix. The user u_i's ratings on the item list $\left\{ I_1^{(u)}, \ I_2^{(u)}, \ldots, \ I_{N_v}^{(u)} \right\}$ are equivalent to the $\mathcal{E}$ between node $\mathcal{U}_i$ and node list $\mathcal{V}_i, \mathcal{V}_2, \ldots, \ \mathcal{V}_{N_v}$. In other words, the type of links on the predicted bipartite graph is equal to the type of rating users will predict for the item.

3 Social Modeling for RS

In this section, we will describe our social modeling methods.

3.1 Opportunities and challenges of social modeling

3.1.1 Oppertunities

The social recommendation is a recommendation system that incorporates social information. Compared with other types of recommendation systems, the biggest difference lies in its consideration of the relationship between users. In recent years, the popularity of online social platforms enables researchers to obtain a large amount of social relationship data and analyze its benefits to recommendation systems.

Considering the influence of friends on the recommendation system, it is more intuitive, and it is in line with life experience. In addition, using social data as a supplement to recommendation system can alleviate the problem of sparse rating data. In fact, for a long time, the analysis of social relations has been an important research content in sociology and complex network science. The recommendation system also hopes to enhance the business value and user experience of the platform by mining the hidden information behind social relationships.

3.1.2 Challenges

Although the social recommendation system has great potential, it is still suffers form some factors. There is a lot of noise in the obtained data of social relations. For instance, in online social networks, users do not necessarily establish contact because of similar interests or preferences. Therefore, if we directly trust the data of this social relationship, it may have a negative impact on the accuracy of recommendation.

On the other hand, on websites with social attributes, our selection and evaluation of something will be influenced by high-impact users, who generally have many followers, whose opinions or ratings are more influential and persuasive, and can be recognized by a large number of users, thus producing similar views.

Based on the above knowledge, this paper attempts to re-model the social influence from the perspective of information dissemination, and apply it to the subsequent recommendation system.

3.2 Spread the influence through a "Ground Node"

Different from most social recommendation algorithms based on propagation between neighboring nodes, our method pay extra attention to the influence of high-impact users on the others in the network. Therefore, an important task in our social modeling method is to find users with high influence. Here, we refer to some theories in sociology to explore the key opinion leaders, i.e. LeaderRank Lü et al (2011), and define our social influence model.

3.2.1 Finding high influence users

For a given original directed network $\mathcal{G} = (n, \ m)$ with n nodes and m edges, which nodes are not necessarily connected. Add a "**Ground Node**" and establish bidirectional connections between this node and all other nodes in the original network, as shown in the figure 2. At this time $\mathcal{G}$ becomes $(n + 1, \ m + 2n)$ with $n + 1$ nodes and $m + 2n$ edges.

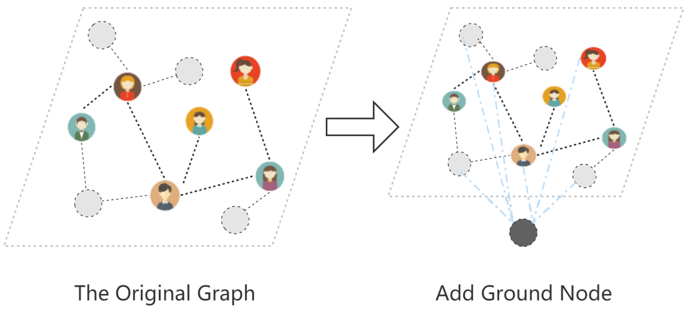


Fig. 2 Add the Ground Node to the original social network (graph).

From the point of view of calculation graph, the addition of "Ground Node" brings two main gains. **Firstly**, the original graph is now converted into a connected graph, thus ensuring the convergence of the following propagation algorithm. **Secondly**, the addition of "Ground Node" also reduced the radius of the network and improved the convergence speed.

Besides, from the perspective of socail modeling, the addition of "Ground Node" also has some pratical significance. As a medium of global influence, this node has the function that people who have no contact with you can also influence you, such as posters and stars in subway advertisements, which is different from previous modes that social influence jumps according to user contact.

The formula for propagation is defined as follows:

$$s_i(t+1) = \sum_{j=1}^{n+1} \frac{a_{ji}}{k_j^{out}} s_j(t), \tag{2}$$

$$s_i = s_i(t_c) + \frac{s_g(t_c)}{n}. \tag{3}$$

Where n represents the total number of nodes in the network; a_{ji} indicates whether node j has a connection to node i, if so, $a_{ji} = 1$, otherwise $a_{ji} = 0$, k_j^{out} represents the number of edges that node j points to other nodes; $s_i(t)$ represents the score of node i at time t; t_c represents the time when $s_i(t)$ converges; $s_g(t_c)$ represents the score of the "Ground Node" at the time t_c; S_i represents the final score of node i. At the beginning of the algorithm, the influence of the "Ground node" is assigned 0, and the other nodes are assigned 1, which is iterated until convergence, that is, the influence value of each node is at a relatively stable value.

3.2.2 Re-quantify the influence of users.

In the previous part, we got the preliminary influence values, but these values are highly discrete and need further processing. First, we sort them from high to low according to the influence value. And in this paper, we use l_i to express the ranking of the user node u_i, that is when $l_i = 1$, it means u_i ranks first in influence. Then, we define a normalization function $\mathcal{F}$ to recalculate the influence values w_i according to the ranking. The function $\mathcal{F}$ is defined as follows:

$$w_i = \mathcal{F}(l_i) = \frac{1}{1 + \log l_i}. \tag{4}$$

The decreasing function $\mathcal{F}$ makes the user's influence value w_i fall in the interval (0, 1), and the top-ranked users have a high influence value.

In this part, we re-quantify the influence of users in social networks. In the next section, we will combine the influence model with graph autoencoder model to serve the rating prediction task together.

4 Proposed approach

In this section, we present the **S**ocial **I**nfluence Modeling Integrating **G**raph **A**utoencoder (SIGA) method. We start with the overall architecture of the model, and then focus on GAE recommender and how it combines with social modeling defined in section 3.

4.1 Complete model framework

Our complete rating prediction model is a hybrid structure of GAE network and social influence model. Fig. 3 shows the schematic diagram of the model.

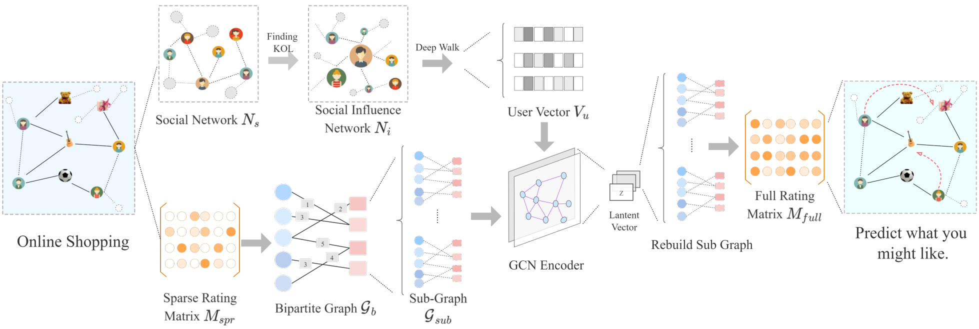


Fig. 3 The Schematic Diagram of SIGA.

In Fig. 3, the original input is a heterogeneous online shopping data set, which is then divided into two paths. First, a sparse rating matrix M_{spr} is established by using the interaction between users and commodities. Second, according to the attention relationship among users in the data set, the social network N_s is established. In the first part, the obtained rating matrix is transfered to a bipartite graph $\mathcal{G}_b$, and then the original bipartite graph is divided into different subgraphs $\mathcal{G}_{sub}$ by different rating levels. In the path of social network, add a "Ground Node", find key opinion leaders, and rebuild a social influence network N_i. Then, in N_i, according to the different influences of users, a large number of wandering sequences in social networks are generated. Then, with the help of the word2vec method, the influence-based users vector representation group V_u is obtained.

Next, the above two paths will be merged. The user vector group V_u and the bipartite subgraph $\mathcal{G}_{sub}$ are taken as the inputs of graph autoencoder model with two-layer GCN encoder. In GAE model, the input is encoded into a hidden vector group, and then they are decoded by a decoder to reconstruct new subgraphs. After each subgraph is processed separately, the new subgraphs are aggregated to obtain a full rating matrix M_{full}. The detailed aggregation method will be introduced in the following section. At this point, we get the prediction of each user's preference for each item.

4.2 A graph autoencoder (GAE) based recommender

In this section, we will introduce the basic recommender GAE, which is based on Graph Neural Network (GNN) and Autoencoder (AE) technology.

4.2.1 Why we use GAE

GNN is a kind of neural network used on graph structure, and have been proved to be effective in capturing and characterizing nodes in the graph structure. Graph Convolution Neural Networks (GCN) are methods derived from traditional Convolutional Neural Network, which is mainly suitable for semi-supervised learning tasks and is the foundation of many other graphic neural networks. Another method with widely applicability is Graph Autoencoder, which is an extended application of Autoencoder (AE) on graphs. The traditional AE is naturally suitable for the recommendation system task of rating prediction. AE is an unsupervised model, and its basic objective function is to minimize the reconstruction error of input and output, which coincides with the rating prediction of the recommendation system. On the bipartite graph, the target of the GAE is to reconstruct the connected edge that representing ratings. Besides, the encoder and decoder of AE can be composed of neural network architecture with arbitrary structure. Therefore, We adopt the GAE network with GCN as the base recommender in our work since it is powerful and suitable for our task.

4.2.2 Encode nodes with GAE

For a given bipartite graph constructed from a user-item rating matrix M, from the item node $\mathcal{V}_j$ to the user node $\mathcal{U}_i$ can be expressed as hidden state vector by means of the feature extractor GCN. This link extractor operation is a kind of local graph convolution, which can be regarded as message transmission on the bipartite graph, that is, vectorized messages are transmitted and transformed on different edges of the graph. Showing as below:

$$\mu_{j \to i} = \frac{1}{C_{ij}} \mathcal{W} x_j^v, \tag{5}$$

where the C_{ij} is the regularization term, $\mathcal{W}$ is the parameter matrix, and x_j^v is the initial feature vector of node $\mathcal{V}_j$. For C_{ij}, $\mid \mathcal{N}_i \mid$ or $\sqrt{\mid \mathcal{N}_i \mid\mid \mathcal{N}_j \mid}$ is usually chosen, which represent left regularization and normalized regularization respectively. $\mid \mathcal{N}_i \mid$ is the number of neighbor nodes of node i. For x_j^v, if there is no extra feature, it is usually replaced by a unique one-hot vector. $\mathcal{W}_r$ is used instead of $\mathcal{W}$ when dividing a graph into subgraph according to different ratings, at this time, formula 5 becomes:

$$\mu_{j \to i,\ r} = \frac{1}{C_{ij}} \mathcal{W}_r x_j^v. \tag{6}$$

Besides the message transmission from items to users, there are also message transmitted from users to items, which are almost equivalent in form.

Take $\mu_{j \to i}$ as an example, the local graph convolution here is equivalent to transferring information x_j^v from item node $\mathcal{V}_j$ to the user node $\mathcal{U}_i$. It was mentioned before that bipartite graph is divided into serveral subgraphs by different rating levels r, so the information transmitted here is also performed separately on each subgraph. The information transmitted by all the neighboring nodes of $\mathcal{U}_i$ is accumulated, as shown in Formula 7:

$$\sum_{j \in \mathcal{N}_r(u_i)} \mu_{j \to i,r}, \ for\ r\ in\ \{1, 2, \ldots, R\}. \tag{7}$$

Then, these representations collected from different subgraphs need to be aggregated:

$$h_i^v = \sigma \left[\text{accum} \left(\sum_{j \in \mathcal{N}_1(u_i)} \mu_{j \to i,1}, \ldots, \ldots, \sum_{j \in \mathcal{N}_R(u_i)} \mu_{j \to i,R} \right) \right], \tag{8}$$

accum in the Formula 8 is a convergence operation, for example, a connection operation or a summation operation can be adopted, and σ is an activation function, such as *ReLU* used here. A complete Formula 8 is called a graph convolution operation, and multiple graph convolution layers can be superimposed. And in order to get the final embedding of $\mathcal{U}_i$, it is necessary to add a dense layer for transformation:

$$z_i^u = \sigma\left(\mathcal{W} h_i^u\right). \tag{9}$$

This part belongs to the feature extraction of GAE, and z_i^u is the middle layer hidden vector. Then in the next subsubsection, we will introduce the decoding process of GAE.

4.2.3 Prediction and GAE training

In order to reconstruct the bipartite graph, we refer to the past experience and use a bilinear decoder. Different rating levels are regarded as one category. The reconstruction of the bipartite graph has become a classification problem for predicting the connected edge types. Bilinear operation is followed by softmax function, which is used to produce the probability of each category:

$$p(\hat{M}_{ij} = r) = \text{softmax}((z_i^u)^T Q_r z_j^v). \tag{10}$$

Q_r is a trainable specific parameter matrix when the edge type is r. The final rating is the expectation about the above probability distribution:

$$\hat{M}_{ij} = g\left(u_i,\ v_j\right) = \mathbb{E}_{p\left(\hat{M}_{ij}=r\right)}\left[r\right] = \sum_{r\ \in\ R} r\ p\left(\hat{M}_{ij} = r\right). \tag{11}$$

For model training, the goal of optimization is to maximize the probability that the prediction ratings are consistent with the true value. There, a method that minimizes the negative log-likelihood of the predicted ratings $\hat{M}_{ij}$ is used:

$$\mathcal{L} = -\sum_{i,j;\ \Omega_{ij}=1} \sum_{r=1}^{R} I\left[M_{ij} = r\right] \log p\left(\hat{M}_{ij} = r\right). \tag{12}$$

Only when $M_{ij} = r$, $I[M_{ij} = r]$ is equal to 1, otherwise $I[M_{ij} = r]$ is equal to 0, this makes it possible to calculate only the logarithmic loss of the true rating. $\Omega \in \{0,\ 1\}^{N_u\ \times N_v}$ represent the mask of the rating, if the value is 1, it represents the observed rating, and if it is 0, it corresponds to the unobserved rating. And only the observed rating level needs to be optimized.

The whole process of encoder, decoder and tarining can be summarized as follows: the embedded representation of users and items is obtained by the encoder (formula 9). Then the probability of the connected edge type is calculated by the decoder (formula 10), and the maximum expectation is selected as the rating prediction value. Finally, according to the cross-entropy loss function (Formula 12) of the rating multi-classification problem, the network is optimized. When making specific predictions, use Formula 11 to predict the rating.

4.3 Combine Social Modeling with GAE

The GAE recommender integrates the interaction between users and items into their latent vector respectively, which is powerful for capturing the features between users and items, and also brings a good explanation to the recommendation system (will be proved in detail in the experimental part). Although the GAE-based method has incorporated the indirect user-to-user relationship through the secondary transmission on bipartite graph, it still can not fully reflect the influence of one user on the purchase of other users in the community. On the other hand, understanding the detailed user interests at the attribute level is also helpful to improve the interpretability and performance of the recommender. Therefore, our idea is to combine the social influence information mined from user connections into GAE to enhance the recommendation.

4.3.1 Details of combination

In section 3, we have quantified the user influence in social networks. Now, we are going to integrate the above-mentioned GAE with social influence model. The key here is how to deal with the structure of integrating the two parts. If we start from the perspective of modifying the social network model, we will find that social influence can be input into GAE model by transforming it into vector group, so some classical embedding methods from sequence to vector will be effective, such as DeepWalk Perozzi et al (2014) we used here. DeepWalk algorithm mainly includes two steps, the first step is random walk sampling node sequence, and the second step is using skip-gram model word2vec to learn expression vector. In the algorithm flow of DeepWalk, it is necessary to formalize the probability of jumping from node u_i to its adjacent node u_j after reaching the node u_i. The probability of jumping from node u_i to u_j can be defined as for formula 13:

$$\mathcal{P}\left(u_j \mid u_i\right)=\begin{cases}\frac{w_j}{\sum_{k\in\mathcal{N}\left(u_i\right)} w_k}, & v_k \in \mathcal{N}\left(u_i\right) \\ 0, & e_{ij} \notin \varepsilon\end{cases}. \tag{13}$$

Where the ε is the set of all users' edges of the social network, $\mathcal{N}\left(u_i\right)$ is the set of all neighbors of u_i, and w_j is the influence of user node u_j. Therefore, the jumping probability between nodes is the ratio of the influence of the target node to the sum of the influences of all neighboring nodes.

After defining the jumping probability of random walking, we also need to specify the depth of the jumping path, and then start to choose the starting point at random. Each walk starts from the selected node, selects a node from the nodes adjacent to the current node according to probability, moves to the node, and repeats this process until the designated jump depth is reached. Once the jumping experiments have been completed, we got a series of sequences, which includes the characteristics of communication between users. The jumping operation makes it possible to reflect the influence between users in real society. For a sequence, each node is regarded as a word, and the sequence can be analogized into a sentence. Then, the vector representation of the user node is trained according to the word2vec method.

What needs special attention here is that we used a "Ground Node" when defining influence propagation, but when generating user vector representation, the "Ground Node" here is forbidden, which is also consistent with real life. Ordinary users will be affected by high influence users in various social media and commercial advertisements, but these users usually only affect their surrounding users, so they can't jump backward with the help of "Ground Node", so when generating user vector, they still jump according to local nodes.

5 Experimental evaluation

This section consists of three subsections, including settings for the experiments, comparison of results, and interpretable analysis of the model.

5.1 Experiment Setup

5.1.1 Construction of the datasets

In order to measure the performance of our proposed model, we use four real world recommendation data sets from different fields, namely Douban Monti et al (2017), FilxSter Monti et al (2017), FilmTrust Golbeck et al (2006), CiaoDVD Lin and Cohen (2010).

Douban is a Chinese literature website, and Douban data set provides users with ratings and comments on books and movies. Douban users can rate items with a score of 1 to 5.

Flixster is a social movie website, which allows users to share movie ratings, discover new movies, and meet other people with similar movie tastes. Users can score movies between 0.5 and 5 points. Like the Douban dataset, the FilxSter dataset also contains bi-directional friend data, that is, if user A is a friend of user B, then user B is also a friend of user A.

FilmTrust is a website for sharing and exchanging movies. FilmTrust data set is a small data set captured from the FilmTrust website in June, 2011. It contains the rating information of users on movies and social information among users, where the rating ranges from 0.5 to 4, with an interval of 0.5.

CiaoDVD is a dataset crawled from the entire category of DVDs from the dvd.ciao.co.uk website in December 2013. Users can also score 1 to 5 points on selected DVDs.

We summarize the detailed statistics of the datasets in Table 1.

Table 1 Statistics of Datasets

Datasets	Users	Items	Ratings	Scale	Density	Relations
Douban	3000	3000	136891	[1, 5]	1.52%	2688
Flixster	3000	3000	26173	[0.5, 5]	0.29%	59354
FilmTrust	1508	2071	35497	[0.5, 4]	1.14%	1853
CiaoDVD	7375	99746	278483	[1, 5]	0.0379%	111781

5.1.2 Task setting

In all the experiments, for our model SIGA, dropout rate is 0.7, the optimizer is Adam with a learning rate of 0.01, the activation function of graph convolution is ReLU, and the attenuation factor of EMA (Exponentially Moving Average) for model smoothing is set to 0.995. And we divide the rating records into two parts with a ratio of 8: 2, namely training set and validation set. SIGA is training and optimized on the training set. As the model in this paper is a generation model, it will generate the ratings of all users for all items. Here we use an evaluation index widely used in previous work in the field of rating prediction: RMSE (Root Mean Square Error) to evaluate the prediction accuracy of the proposed method and other baseline methods. It measures the deviation between predicted and actual value, so the lower the value, the better. The following is the formula of the evaluation indicator:

$$RMSE = \sqrt{\frac{\sum_{r_{ij} \in R_{test}} \left(r_{ij} - \hat{r}_{ij}\right)^2}{\mid M \mid}}. \tag{14}$$

5.1.3 Methods to compare

Because this article focuses on the task of rating prediction, many top-K methods are not suitable for comparison. In this part, we consider the following methods for performance comparison:

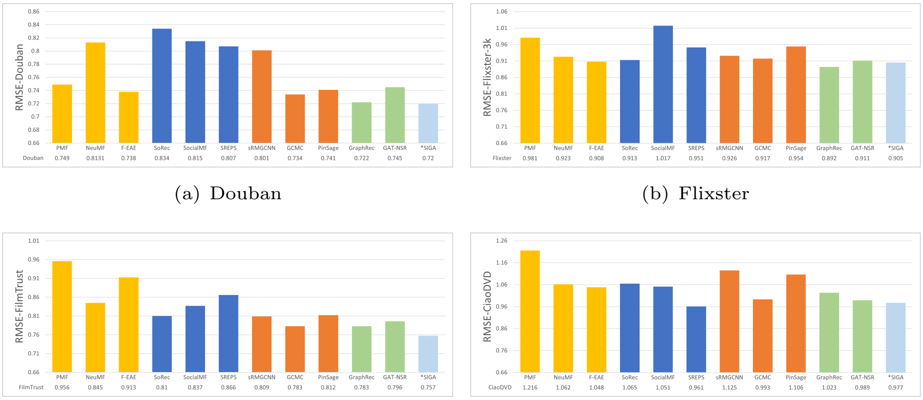
- PMF Paatero and Tapper (1994): It is a widely used probability matrix factorization algorithm, which does not consider any social relationship data.
- NeuMF He et al (2017): A classic and effective solution of collaborative filtering based on neural network.
- F-EAE Hartford et al (2018): It uses exchangeable matrix layers to perform inductive matrix completion without using content.
- SoRec Ma et al (2008): It is a model based on a probability matrix and a social recommendation algorithm that combines a user-item rating matrix and user social relationship at the same time.
- SocialMF Jamali and Ester (2010): This is a social recommendation method that makes the embedding vector of the user close to the vector of his direct friends.
- SREPS Liu et al (2018): It proposes a collaborative algorithm based on essential preference space, and integrates explicit feedback, implicit feedback and social relations.
- sRMGCNN Monti et al (2017): It uses graph convolution on the nearest neighbor graph of users and items and uses RNN to learn representation iteratively.
- GCMC Berg et al (2017): A state-of-art rating prediction model based on graph autoencoder, this autoencoder generates the hidden features between user and item by means of information transmission in a bipartite interaction graph.
- PinSage Ying et al (2018): It is the first successful application of GCN in a business recommendation system, and it is an inductive model based on node-level GNN using content.
- GraphRec Fan et al (2019): This method is a graph neural network sechema for social recommendation, which models the user-user graph and user-item graph respectively and then serves the rating prediction task.
- GAT-NSR Mu et al (2019): It combines graph attention network with social recommendation.

Our baselines cover the four related directions of the rating prediction models. Among them, PMF, NeuMF, F-EAE are machine learning methods which do not contain extra information. SoRec, SocialMF and SREPS make additional use of social information. sRMGCNN, GCMC and PinSage are methods based on graph neural network, which do not contain extra social relationships. GraphRec and GAT-NSR are also graph-based methods, but the difference is that they take advantage of social relations. Parameters of these models are referred to the corresponding original papers.

5.2 Result and analysis

In this subsection, firstly, we will compare the performanace of all the above methods and analyze the results. Then, we will carry out hyperparameter verfication and ablation experiments on our proposed methods.

5.2.1 Performance comparison



(a) Douban (b) Flixster

(c) FilmTrust (d) CiaoDVD

Fig. 4 The experimental results on four data sets, comparison methods are marked with the same color in groups, and the method in this paper is marked in blue. The the indicator lower is better.

Fig. 4 shows the overall prediction error i.e. RMSE of all methods on four data sets. The algorithms are classified into four groups, namely pure machine learning group, social information group, graph neural network group and social information combined graph neural network group. The findings from the figure are briefly described as follows:

- In terms of average performance on four data sets, Graph neural network group is superior to social network group and pure machine learning group. This proves the effectiveness of graph neural network in feature extraction.
- The social information combined graph neural network group is superior to the other three groups, which shows that the combination of social network and graph neural network has a gain effect.
- Except on Douban data set, the overall performance of social information group is better than that of pure machine learning group. The small amount of social relations data in Douban data set may be one reason.
- Except on FilmTrust data set, F-EAE always perform better than PMF, NueMF, SoRec, SocialMF. All of these methods are based on matrix factorization. The difference is that F-EAE is an inductive method. This shows that induction is of great significance in matrix factorization model.
- GCMC is a powerful matrix completion method, which is obviously superior to all models in the other three groups. Because its bipartite graph information transmission mode is naturely suitable for extracting the structural information of the graph.
- In the baseline method, GraphRec and GAT-NSR are both based on graph neural network and social information. GraphRec combines scoring

information with user-item graph through a new model structure, while GAT-NSR combines graph attention network with social information. They are significantly better than other methods.
- Our method SIGA consistently outperforms all the baseline methods. Compare with GCMC, ourmodel provides advanced model components to make used of social influence. Compared with GraphRec and GAT-NSR, our modeling method combining graph autoencoder and social influence is more effective, and it is also very suitable for rating prediction tasks. In the next subsubsection, we will provide further analysis to better understand the contribution of model components to the model effect.

5.2.2 Ablation experiments and superparameter verification

In this subsubsection, we will verify the effectiveness of the main parts of the model and the hyperparameters of the model.

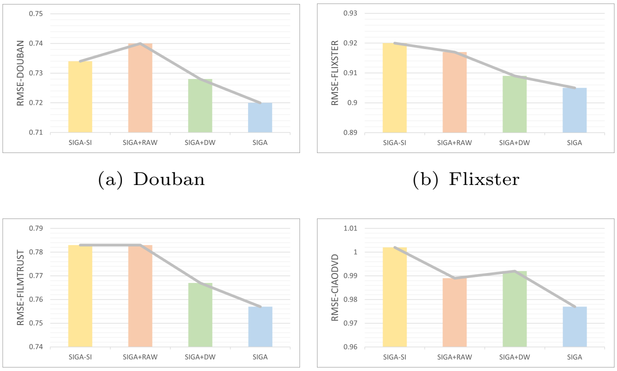
A. Effect of Soical Modeling. In the preivous section, we demonstrated the performance of our model. Our model includes two main components, social modeling and graph autoencoder. To understand the working of social modeling component, we will compare SIGA with three variants: SIGA-SI, SIGA+RAW, SIGA+SN. Details of these variants are shown below.

- SIGA-SI: On the basis of original SIGA model, the social influence is removed. All users in GAE are given the same initial value.
- SIGA+RAW: Here, we tend to use the original social relationship. However, because the number of users is high, the expression of One-Hot vector is too long. We let the input connect to an MLP layer and compress it into a 128-dimensional vector.
- SIGA+DW: For the original user social relationship data, build a social relationship graph. Then, the user's vector representation is directly generated by deepwalk. We do several operations to get the average value.

Here, we focus on analyzing the effectiveness of our social modelling method. The performance on four data sets are shown in Fig. 5. Among the three comparison methods, SIGA-SI method is the worst because it does not use social information. Compared with SIGA-SI, SIGA+RAW, which directly uses social information, has not been improved much. Compared with the first two methods, SIGA+DW has a certain degree of improvement, but the effect is not stable, because social relationship data can not directly reflect users's preferences, and relying too much on this information may have negative effects.

B. Effect of graph autoencoder. To verify the effect of the GAE part of our model, we replaced the GAE feature extractor in SIGA with the classic general model NueMF. The user representation obtained by social modeling is used as the input of NueMF. We compare SIGA with classicial NueMF and NeuMF+SI, details are given below:

- NeuMF: Basic neural matrix decomposition model, the input is user and item vector.



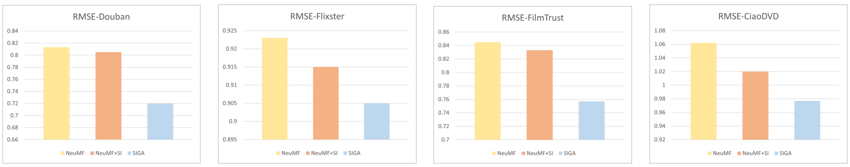
(a) Douban (b) Flixster

(c) FilmTrust (d) CiaoDVD

Fig. 5 Ablation experiment on social modeling. Compare the performance of RMSE indicators of the model on four data sets, lower is better.

- NeuMF+SI: On the basis of NeuMF, keep the social modeling part of SIGA, and then replace the original input of NeuMF with the obtained user representation.

Here, we focus on analyzing the effectiveness of our GAE component. The performance on four data sets are shown in Fig. 6. It can be seen from the figure that after replacing the GAE module in SIGA with the classic NeuMF model, the performance has declined. However, with the addition of Socail Influence, the original NeuMF has been improved.

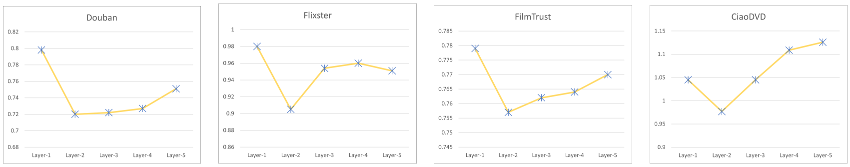


(a) Douban (b) Flixster (c) FilmTrust (d) CiaoDVD

Fig. 6 Ablation experiment on GAE component. Compare the performance of RMSE indicators of the model on four data sets, lower is better.

C. Superparameter verification. In this part, we briefly compare the influence of the number of GCN layers in SIGA on the model effect. According to the famous "Six Degrees of Separation" theory, no more than five people will be separated from any stranger. And the GCN layers stand for information transfer from user to item to user, so we consider comparing the effects of GCN layers from layers 1 to layer 5. Results are shown in Fig. 7.

By observing Fig. 7, we can find that when the number of GCN layers is two, there will be better results. On Flixster and CiaoDVD data sets, with the



(a) Douban (b) Flixster (c) FilmTrust (d) CiaoDVD

Fig. 7 Comparison of the model with different GCN layers.

increase of layers, the effect of the model will deteriorate earlier. We speculate that this is because their data is more sparse than Douban and FilmTrust data sets, and the GCN of the two layers has been able to generate enough interactions in the local area of users, too many layers will lead to over smoothing. On the other hand, the increase of the number of layers will lead to more calculations. To sum up, this paper has chosen two levels of GCN.

5.2.3 Summary

In the above experiments, in most cases, our method is superior to other baseline methods. Compared with social recommendation algorithms, our model has a significant improvement. On the other hand, compared with the current advanced graph-based recommendation algorithms, our model has made a breakthrough after incorporating social influence. This discovery shows that if social relations are handled properly, they can show amazing results in rating prediction tasks. Then, in the next subsection, we will analyze the interpretability of the model.

5.3 Interpretability of Recommendation Model

Previous experiments show that our SIGA model can produce high-quality rating predictions. Another major advantage of SIGA is that the model is well interpretable. Recall that we use $\mathcal{F}(l_i)$ to compute the weights over the users in Formula 4. Our model can produce a distribution of neighbors' weights for user u. And the latent user representation x^u is highly correlated with the weights of neighbors. The neighbors' weights provide an attribute-level interpretation. Furthermore, the specific message passing mechanism on bipartite graph structure maintains the collaborative signals, which further provides a structure-level interpretation. To facilitate understanding, we present an example in Fig. 8.

5.3.1 Attribute-level interpretability

Fig. 8 presents an interaction of users from a small community of six people. Suppose it is already known some users play a key role in decision-making. First, node 1 in the graph is randomly selected as the starting point, and the nodes adjacent to 1 are 2 and 5. At this time, the next jump probability is calculated according to the influence weights of nodes 2 and 5. Then, it jumps to node 2, which can only go to node 3, and then node 3 jumps to node 1 or node 4 according to weight probability. Assuming that we limit the jump length to 5, we get a user sequence of 1, 2, 3, 4, 6, and 1 according to the path shown in the figure. Randomly select the starting point and repeat it several times in the above way, a series of user sequences are generated. The higher the influence of user nodes, the more times it appears in the sequence. User groups with high correlation have high co-occurrence times in the sequence. Finally, the user vector representation is obtained by using the word2vec method on these sequences. And it is clear that the user vectors obtained by our model have higher weights for users with higher influence, and the associated user vectors are closer in space. The information transmission in the community is dynamic, and our model can capture the evolving spread preference according to the influence attributes.

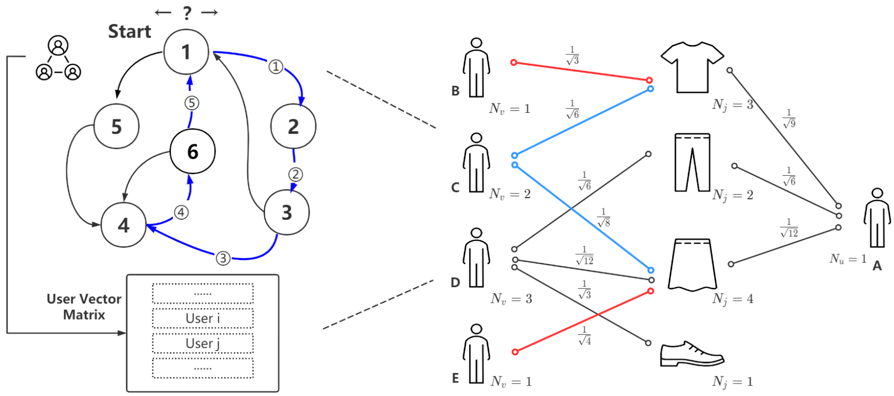


Fig. 8 Schematic diagram of interaction in a small user community. Dark blue indicates the skip path in the community. The left side of the figure shows the process of obtaining the user vector matrix from the social network with KOLs. The right part shows the disassembly of message transmission in a bipartite graph. We highlight the example connection with red and sky blue. The number on the side of each link represents the weight of influence. The small mark on the side of each user (item) refers to the number of directly related items (users).

5.3.2 Structure-level interpretability

Inspired by lightGCN He et al (2020), we found that our model has the effect of collaborative filtering after some simplification. Taking GAE with two-layer feature extractors as an example, different kinds of rating r and nonlinear activation function σ are ignored here. For convenience of understanding, let $C_{ij} = \sqrt{|\mathcal{N}_i||\mathcal{N}_j|}$ and $\mathcal{W}x_j^v = h_j$ in formula 5. Formula 8 represents the process of feature extraction, which is essentially an aggregation operation between neighboring nodes. Next, we will briefly prove how the collaborative filtering effect is produced between user u and user v:

$$h_u^{(2)} = \sum_{j \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_j|}} h_j^{(1)} \tag{15}$$

$$= \sum_{j \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_j|}} \left(\sum_{u \in \mathcal{N}_j} \frac{1}{\sqrt{|\mathcal{N}_j||\mathcal{N}_v|}} h_v^{(0)} \right) \tag{16}$$

$$= \sum_{j \in \mathcal{N}_u} \frac{1}{|\mathcal{N}_j|} \sum_{v \in \mathcal{N}_j} \frac{1}{\sqrt{|\mathcal{N}_u|| \ mathcal N_v|}} h_v^{(0)} \tag{17}$$

u and v represent the user nodes on the left side of the bipartite graph, j represents the item node on the right side, and h is the representation of each node. The superscript of h represents the embedding representation of which layer. Here, we deduce from the second layer. After a series of derivation, formula 15 is obtained. User u can be expressed by the users who have interaction records with him, and the neighbor number $|\mathcal{N}_j|$ of item j is only a weight coefficient. Furthermore, items without indirect interaction do not affect the above expression, therefore formula 17 is further simplified to:

$$h_u = \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_v|}} \sum_{j \in \mathcal{N}_u \cap \mathcal{N}_v} \frac{1}{|\mathcal{N}_j|} h_v. \tag{18}$$

By analyzing formula 18, we can see that it has strong interpretability: the more items of common interest between two users (the more sum items there are), the greater the influence of v on u; the less popular an item of common interest is, the higher its influence will be, corresponding to $\frac{1}{|N_j|}$ here; the less interactive behavior of v, the greater the value of $\frac{1}{\sqrt{|\mathcal{N}_v|}}$, which means that v has a greater impact on u. This has something in common with user-based collaborative filtering methods.

An example: Although we have proved the collaborative filtering characteristics of our graph model by using formulas (15-18), can we further predict how the user will select a set of items according to the associated users? The right part of Fig. 8 shows the message delivery process from user to user (item to item is the same). The values on the lines in the figure are used to indicate the degree of influence. For convenience, we call user A as u, other users as v, and items as j. User vectors are represented by handwritten letters, such as $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$. Taking these five users and four items as an example, we will find out how user A's decision is influenced by other users.

Take the user B as an example, user B only has interaction records with T-shirt, so $N_v = 1$, it means user B has one neighbor. While the T-shirt has been purchased by three users B, C, and A, $N_j = 3$. Therefore, according to Formula 15, the correlation between the T-shirt and user B is $\frac{1}{\sqrt{1*3}}\mathcal{B}$. According to formula 16, the T-shirt can be expressed by users B, C, and A. Then, the association between users C and A and the T-shirt is calculated respectively, and the new expression $\frac{1}{\sqrt{3*1}}\mathcal{B} + \frac{1}{\sqrt{3*2}}\mathcal{C} + \frac{1}{\sqrt{3*3}}\mathcal{A}$ of the T-shirt can be obtained by summarizing them. By analogy, a new representation of all clothes related to user A is obtained, and the transmission from the item node to the user node is carried out again. Finally, user A can be expressed as $\left(\frac{1}{9}+\frac{1}{6}+\frac{1}{12}\right)\mathcal{A}+\frac{1}{3\sqrt{3}}\mathcal{B}+\left(\frac{1}{3\sqrt{6}}+\frac{1}{4\sqrt{6}}\right)\mathcal{C}+\left(\frac{1}{6}+\frac{1}{12}\right)\mathcal{D}+\frac{1}{4\sqrt{3}}\mathcal{E}$. For the convenience of intuitive feeling, it is converted into decimal representation and the first three digits are reserved, that is $0.361\ \mathcal{A}\ +0.192\ \mathcal{B}+0.238\ \mathcal{C}+0.25\ \mathcal{D}+0.1444\ \mathcal{E}$. Compare the expression with Fig. 8, besides user A, user D has the greatest influence on user A, while user E has the smallest influence weight. Let's think about why.

Taking user B and user E as examples, B has a higher influence than E. B and E both have only one interaction record associated with user A, refer to the red line in the figure. The difference is that skirts are more popular than T-shirts and the less popular items contain more information. An item is not popular, but both u and v have interacted, so the better collaborative filtering effect produced by this item is obvious.

Then, observe users B and C, they both bought the T-shirt, but the difference is that user C also bought a skirt. The purchase connection of user C is marked with sky blue in the figure. The influence weight of user C is higher than that of user B because there are more items of common interest between C and A.

On the other hand, the influence of user C on user A is transmitted through two paths. In the path of the T-shirt, the influence of user C is less than that of user B. Because of the less the interaction behavior of v, the greater the influence of v on u. The more inactive users provide collaborative signals, the more important they are. u and a very inactive user v have interacted with an item together. Obviously, this is a rare situation, and v can provide more collaborative signals at this time.

At this point, we have a rational understanding of the interpretability of the model structure.

6 Related Work

Our work is closely related to the following research directions.

Social Recommendation A long time ago, researchers began to introduce social information and make recommendations based on user similarity Basu et al (1998) Massa and Avesani (2007). However, there are many problems with these methods, such as the high sparsity of friend data. Besides, due to the privacy protection of data, many websites with social functions do not disclose their data. This makes the current research mostly based on some small-scale public data sets. In recent years, some researchers have begun to explore implicit social relations, such as social influence, and some good results have been obtained Ling et al (2019) Wu et al (2019a) Liu et al (2020). These studies show that social influence is the direction worth studying.

Graph neural networks (GNNs) Graph neural network is a type of representation learning method based on graph features. One of its core problems lies in how to incorporate graph structure into the machine learning model Zhang et al (2019). There are many kinds of classical graph neural networks, such as Graph Convolution Network (GCN) Kipf and Welling (2016a) He et al (2020), Graph Autoencoder (GAE) Kipf and Welling (2016b), Spatial or Temporal Graph Network (ST-GCN) Yan et al (2018) Ma et al (2020), Graph Attention Network Wu et al (2019b) etc. Among them, the graph convolution network is the foundation of many complex neural network models.

Autoencoders (AEs) User- or Item-based autoencoders Sedhain et al (2015) have been a class of state-of-art collaborative filtering models that can be seen as a special case of graph autoencoder model Kipf and Welling (2016b), and in this case, only the id of users and items are embedded. Auto-Rec proposed by Sedhain et al. Sedhain et al (2015) is a typical model, where all ratings of an item are projected onto a latent space through an encoder layer and then through the decoder layer, the user's rating estimation of the item is obtained. Due to the simple structure of the Auto-Rec model, there is a problem of insufficient expressive ability. In addition to basic auto-encoders, Denoising Autoencoders (DAE) Bengio et al (2013), Variational Autoencoder (VAE) Kipf and Welling (2016b), RecVAE Shenbin et al (2020), MacridVAE Ma et al (2019) etc. have been introduced into the recommendation system, and important progress has been made in many subdivision directions of the recommendation system.

7 Conclusion

In this paper, we proposed a novel graph-based social modeling method SIGA, which integrates the informantion of global influence to remodel social recommendation and server the task of rating prediction. One of the cores of our model is to generate high-quality user latent vector through social modeling, and another core lies in feature extraction on the bipartite graph, which uses a graph convolution encoder to obtain representations of the interaction between users and items. In addition, these two core parts are respectively interpretable in attribute and structure. Extensive experimental results on four real-world datasets demonstrate the effectiveness of the proposed model and the promise of social modeling.

Social networks still have a lot of hidden information worth digging deeper, as future work, we will consider expanding our model by further modeling social networks to improve the recommendation quality. At present, our focus is on the task of rating prediction. We will also study whether our solution can be extended to other complicated tasks.

Acknowledgments. This work is supported by Tianjin "Project + Team" Key Training Project under Grant No. XC202022.

Declarations

Some journals require declarations to be submitted in a standardised format. Please check the Instructions for Authors of the journal to which you are submitting to see if you need to complete this section. If yes, your manuscript must contain the following sections under the heading 'Declarations':

- Funding
- Conflict of interest/Competing interests (check journal-specific guidelines for which heading to use)
- Ethics approval
- Consent to participate
- Consent for publication
- Availability of data and materials
- Code availability
- Authors' contributions

If any of the sections are not relevant to your manuscript, please include the heading and write 'Not applicable' for that section.

Editorial Policies for:

Springer journals and proceedings:
https://www.springer.com/gp/editorial-policies

Nature Portfolio journals:
https://www.nature.com/nature-research/editorial-policies

Scientific Reports:
https://www.nature.com/srep/journal-policies/editorial-policies

BMC journals:
https://www.biomedcentral.com/getpublished/editorial-policies

Appendix A Section title of first appendix

An appendix contains supplementary information that is not an essential part of the text itself but which may be helpful in providing a more comprehensive understanding of the research problem or it is information that is too cumbersome to be included in the body of the paper.

References

Basu C, Hirsh H, Cohen W, et al (1998) Recommendation as classification: Using social and content-based information in recommendation. In: Aaai/iaai, pp 714–720

Bengio Y, Yao L, Alain G, et al (2013) Generalized denoising auto-encoders as generative models. arXiv preprint arXiv:13056663

Berg Rvd, Kipf TN, Welling M (2017) Graph convolutional matrix completion. arXiv preprint arXiv:170602263

Fan W, Ma Y, Li Q, et al (2019) Graph neural networks for social recommendation. In: The World Wide Web Conference, pp 417–426

Golbeck J, Hendler J, et al (2006) Filmtrust: Movie recommendations using trust in web-based social networks. In: Proceedings of the IEEE Consumer communications and networking conference, Citeseer, pp 282–286

Hartford J, Graham D, Leyton-Brown K, et al (2018) Deep models of interactions across sets. In: International Conference on Machine Learning, PMLR, pp 1909–1918

He X, Liao L, Zhang H, et al (2017) Neural collaborative filtering. In: Proceedings of the 26th international conference on world wide web, pp 173–182

He X, Deng K, Wang X, et al (2020) Lightgcn: Simplifying and powering graph convolution network for recommendation. In: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, pp 639–648

Jamali M, Ester M (2010) A matrix factorization technique with trust propagation for recommendation in social networks. In: Proceedings of the fourth ACM conference on Recommender systems, pp 135–142

Kipf TN, Welling M (2016a) Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:160902907

Kipf TN, Welling M (2016b) Variational graph auto-encoders. arXiv preprint arXiv:161107308

Lin F, Cohen W (2010) Advances in social networks analysis and mining (asonam). In: 2010 International Conference on, pp 192–199

Ling Z, Xiao Y, Wang H, et al (2019) Extracting implicit friends from heterogeneous information network for social recommendation. In: Pacific Rim International Conference on Artificial Intelligence, Springer, pp 607–620

Liu CY, Zhou C, Wu J, et al (2018) Social recommendation with an essential preference space. In: Thirty-second AAAI conference on artificial intelligence

Liu Y, Liang C, He X, et al (2020) Modelling high-order social relations for item recommendation. IEEE Transactions on Knowledge and Data Engineering

Lü L, Zhang YC, Yeung CH, et al (2011) Leaders in social networks, the delicious case. PloS one 6(6):e21,202

Ma C, Ma L, Zhang Y, et al (2020) Memory augmented graph neural networks for sequential recommendation. In: Proceedings of the AAAI Conference on

Artificial Intelligence, pp 5045–5052

Ma H, Yang H, Lyu MR, et al (2008) Sorec: social recommendation using probabilistic matrix factorization. In: Proceedings of the 17th ACM conference on Information and knowledge management, pp 931–940

Ma J, Zhou C, Cui P, et al (2019) Learning disentangled representations for recommendation. arXiv preprint arXiv:191014238

Massa P, Avesani P (2007) Trust-aware recommender systems. In: Proceedings of the 2007 ACM conference on Recommender systems, pp 17–24

Monti F, Bronstein MM, Bresson X (2017) Geometric matrix completion with recurrent multi-graph neural networks. arXiv preprint arXiv:170406803

Mu N, Zha D, He Y, et al (2019) Graph attention networks for neural social recommendation. In: 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), IEEE, pp 1320–1327

Paatero P, Tapper U (1994) Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. Environmetrics 5(2):111–126

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

Sedhain S, Menon AK, Sanner S, et al (2015) Autorec: Autoencoders meet collaborative filtering. In: Proceedings of the 24th international conference on World Wide Web, pp 111–112

Shenbin I, Alekseev A, Tutubalina E, et al (2020) Recvae: A new variational autoencoder for top-n recommendations with implicit feedback. In: Proceedings of the 13th International Conference on Web Search and Data Mining, pp 528–536

Wu L, Sun P, Fu Y, et al (2019a) A neural influence diffusion model for social recommendation. In: Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval, pp 235–244

Wu Q, Zhang H, Gao X, et al (2019b) Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. In: The World Wide Web Conference, pp 2091–2102

Yan S, Xiong Y, Lin D (2018) Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Proceedings of the AAAI conference on artificial intelligence

Ying R, He R, Chen K, et al (2018) Graph convolutional neural networks for web-scale recommender systems. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 974–983

Zhang C, Song D, Huang C, et al (2019) Heterogeneous graph neural network. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 793–803