Neural Processing Letters (2024) 56:136
https://doi.org/10.1007/s11063-024-11555-7

SGNNRec: A Scalable Double-Layer Attention-Based Graph Neural Network Recommendation Model

Jing He[1,2] · Le Tang[1] · Dan Tang[1] · Ping Wang[1] · Li Cai[1]

Accepted: 11 February 2024 / Published online: 4 April 2024

Abstract

Due to the information from the multi-relationship graphs is difficult to aggregate, the graph neural network recommendation model focuses on single-relational graphs (e.g., the user-item rating bipartite graph and user-user social relationship graphs). However, existing graph neural network recommendation models have insufficient flexibility. The recommendation accuracy instead decreases when low-quality auxiliary information is aggregated in the recommendation model. This paper proposes a scalable graph neural network recommendation model named SGNNRec. SGNNRec fuse a variety of auxiliary information (e.g., user social information, item tag information and user-item interaction information) beside user-item rating as supplements to solve the problem of data sparsity. A tag cluster-based item-semantic graph method and an apriori algorithm-based user-item interaction graph method are proposed to realize the construction of graph relations. Furthermore, a double-layer attention network is designed to learn the influence of latent factors. Thus, the latent factors are to be optimized to obtain the best recommendation results. Empirical results on real-world datasets verify the effectiveness of our model. SGNNRec can reduce the influence of poor auxiliary information; moreover, with increasing the number of auxiliary information, the model accuracy improves.

Le Tang and Li Cai have contributed equally to this study.

✉ Li Cai
caili@ynu.edu.cn

Jing He
hejing@ynu.edu.cn

Le Tang
tangle4396@mail.ynu.edu.cn

Dan Tang
m15283899275@163.com

Ping Wang
15882350593@163.com

[1] National Pilot School of Software, Yunnan University, Kunmming 650500, Yunnan, China

[2] Engineering Research Center of Cyberspace, Yunnan University, Kunmming 650500, Yunnan, China

Keywords Recommendation system · Graph neural network · Double-layer attention network · Auxiliary information

1 Introduction

With the advent of big data, it is difficult to find content that meets real needs in a massive database [1]. At present, the keyword search provided by the search engines can help people to filter data. However, it cannot reflect the personalized needs of users because the search results are determined by the keywords, which leads to a decline in users' satisfaction with the use of the search engine. Therefore, personalized recommendation has become a topic of considerable research.

The most traditional recommendation algorithm is collaborative filtering, which calculates the nearest neighbors of users or items, and then uses the neighbors for recommendation. The collaborative filtering algorithm has diverse limitations including those caused by data sparsity and cold start; hence, over the years, the algorithm has been improved by many researchers. To solve the problem of data sparsity and cold start, auxiliary information is introduced into the recommendation model. Previous recommendation algorithms mainly used auxiliary information as a supplement to determine users' or items' similarity. For example, many authors [2–4] use tag information as a supplement to calculate similarity in collaborative filtering. The inclusion of tag information can solve the problem of lack of data. To a certain extent, it also improves the accuracy of recommendation because it fully considers the rating relationship between users and items and other aspects of information that traditional collaborative filtering algorithm does not consider. However, these algorithms require complex mathematical modeling and have great difficulties in fusing multi-source auxiliary information.

Traditional algorithms use data in Euclidean space with specific rules. For example, most recommendation algorithms often use the user-item rating matrix, which is used as the eigenvalue matrix to calculate the similarity between vectors. In reality, items that users have not rated should not be taken as eigenvalues. Therefore, there are numerous nulls in the fictitious user-item rating matrix. This causes the problem of data sparsity and leads to low accuracy of the recommendation algorithm. Moreover, most of the data are not in Euclidean space. For example, the rating relationships between users and items resemble the structure of an irregular graph, indicating that different users can score different items. A rating graph contains a disordered node set with a variable size, and each node in the graph has a different number of adjacent nodes, resulting in certain important operations (e.g., aggregation and convolution) that are easy to compute on the graph. As a result, the graph neural network (GNN) and graph convolution neural network (GCN) have attracted considerable research attention in the field of recommendation systems.

The graph neural network recommendation models which fuse various auxiliary information can be divided into two categories: the recommendation based on multi-relationship graphs and the recommendation based on single-relationship graphs.

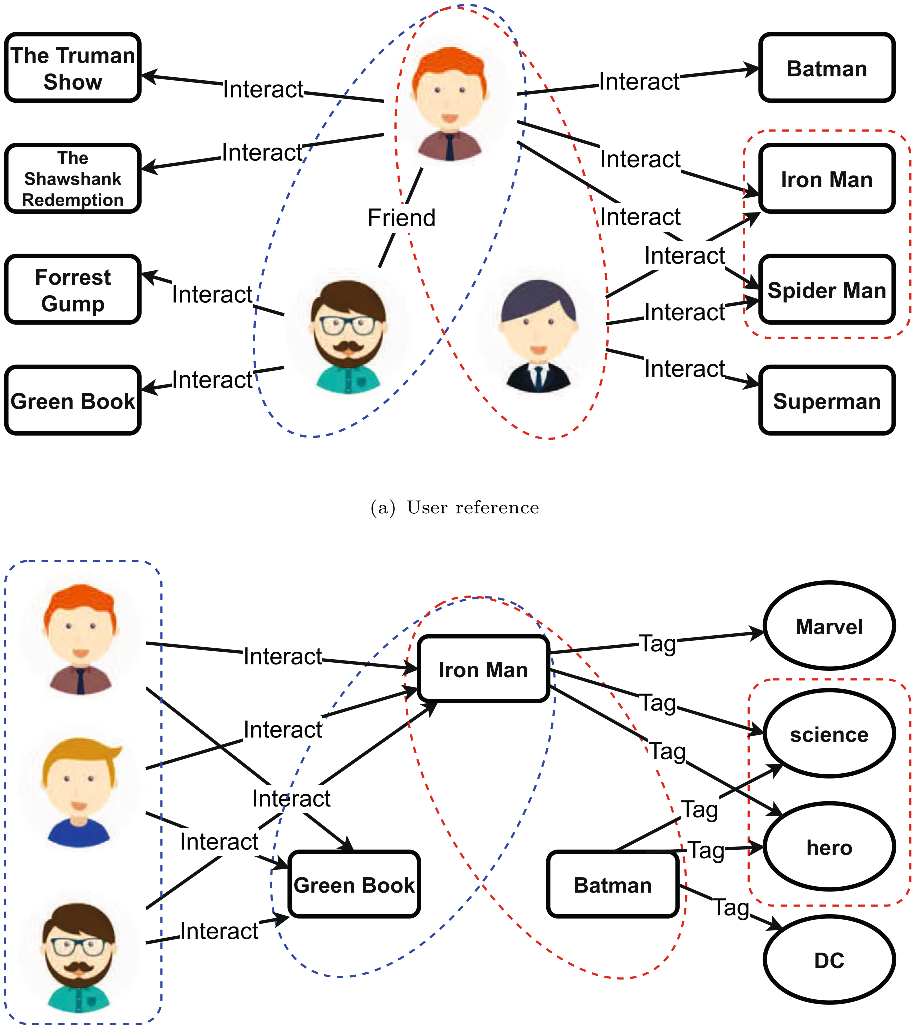
Existing researches [5, 6] adopt multi-relationship graphs recommendation. A multi-relationship graph contains multiple entity relationships. For example, a user-movie graph can represents rating relationship, actor relationship and direct relationship. However, the single aggregation method in the existing learning model can not reflect the differences of multiple entity relationships. This leads to a decrease in the accuracy of the recommendation. Therefore, reference [7] proposed a method to transform multi-relationship graphs into single-relationship graphs.

The Graph Convolutional Matrix Completion (GC-MC) model was proposed [8], which constructed the bipartite graph of user-item rating relationship. Then, the prediction of user-item rating could be obtained by aggregating the information of the neighbor nodes using mean-pooling. However, other auxiliary information except the rating information was not used in this model. Furthermore, the GC-MC model does not consider the difference in the influence of neighbor nodes on users. Then GraphRec [9], a graph neural network recommendation model was proposed. It not only considers the users' rating information but also adds the users' social information when learning the latent factor. Therefore, high-quality recommendation results are obtained in GraphRec. However, auxiliary information is not added to the item model in it. Furthermore, the quality of recommendation results of GraphRec would be low when the quality of social information is poor. Reference [10] indicated that people's preferences are not completely static. Thus, it should not have fixed weights or fixed restrictions. Therefore, social influence should be divided into two forms: dynamic and static. DANSER was proposed, which used dual graph attention networks to collaboratively learn representations for two-fold social effects. At the same time, the social effects of the user domain are extended to the item domain, and the social effects of the item are constructed so as to solve the problem of cold start and data sparsity caused by the lack of item auxiliary information. However, the auxiliary information of the item is also based on the social information of the user rather than the items. Reference [11] proposed a multi-view data method. It is similar to a multi-source graph. However, it does not take into account the different graph relationships have different recommendation weights. Reference [12] proposed DiffNet. It is based on SVD++ and GCN. GCN uses the social network to obtain the user's embedding, and then the SVD++ framework is directly applied. IG-MC [13] used GraphSage in matrix completion. In DiffNet++ [14], which is based on DiffNet, user-item bipartite graph information is added.

Currently, more and more researchers pay attention to the social recommendation. Wei et al. [15] proposed a fusion model of direct friends and indirect friends to discover user's preferences in sparse data scenarios. Moreover, a heterogeneous Trust-based Social Recommendation which integrated implicit neighbors information is designed in [16]. However, they did not fully consider that some implicit social relationships could be noise. Reference [17] proposed a neighborhood denoising method which combines a motif-based GCN and a fully connected multi-layer perception. GNNTSR [18] further considered the reliability of users when assessing the importance of their social information. However, in a recommendation model, the user's social information is only one of the factors, the interaction information between the user and the item, and the characteristic of the item are also essential.

In a word, the existing researches mainly have the following deficiencies:

- The recommendation auxiliary information used by existing models has limitations. For example, in practical applications, ratting information between users and items are often difficult to obtain, which leads to a sparse ratting matrix, resulting in prediction errors. At the same time, as we all know, the interaction between the user and the item also includes the user browse the item repeatedly, add favorites, add to the shopping cart, etc. Adding these information to the recommendation system can effectively improve the accuracy of recommendation.
- On the other hand, existing models do not fully considered the negative effect caused by low-quality auxiliary information. That means, adding too much low-quality auxiliary information in the recommendation will reduce the accuracy of the result.



(a) User reference

(b) Item reference

Fig. 1 Auxiliary information for recommendation

In this paper, the SGNNRec is proposed as an easily extended graph neural network recommendation model fusing multi single-relational graphs. The main contributions of this paper are as follows:

I. An easily extended graph neural network recommendation model fusing multi single-relational graphs is proposed. The model is constructed based on the bipartite graph of user-item ratings, supplemented by isomorphic graphs of other auxiliary information. It can simply and effectively fuse multi-source auxiliary information to solve data sparsity and cold start problems. To prove, the model fuses three kinds of auxiliary information (As shown in the Fig. 1, user social information, item tag information and user-item interaction information).

II. This paper proposes a two-layer attention network structure. The first layer of attention network considers the different strengths of different neighbor information, and the second layer of attention network considers the different qualities of different graph information. It can reduce the influence of low-quality information on recommendation results, so as to increase the accuracy.

2 Fundamentals

2.1 Clustering

As the saying goes, "birds of a feather flock together." Hence, objects are divided on the basis of similarity in different categories; on this basis, multiple clusters are constructed. The objects in the same cluster are highly similar, and the objects in different clusters are diverse.

To cluster objects effectively according to the aforementioned ideas, many algorithms have been proposed, such as K-means [19], DB-scan [20], and spectral clustering [21].

2.2 Word2vec

Word2vec [22] is an important technology in NLP (Natural Language Processing); it is also important for neural networks in general. The main purpose of the technology is to learn the vector representation of different words, such that the vector distance between similarly meaning words is close, and the vector distance between words that are different in meaning is farther. To achieve this aim, it is often needed to establish a fake task model to train the vector representation of these words. These vector representations of words are often a hidden layer in the network model. In the beginning, the purpose of word2vec technology is to get the matrix data of the hidden layer; the forward process of this model does not have much practical value. Hence, the whole model training task is called fake task.

To ensure that the vector distance between words reflects the difference in meaning, it uses the idea that the surrounding words of a word are often close to the word in semantics. According to this idea, two models, CBOW and Skip-Gram, are proposed.

2.3 Graph Neural Network

Perozzi proposed the application of word2vec technology to graph representation learning. Later on, DeepWalk model [23] was proposed in similar lines. With the development of word2vec, item2vec [24] was generated. It shows that not only words can learn to express low dimensional vectors in deep learning, but objects also can do so. Further-more, the nodes of the graph and abstract relationships could be presented by low dimensional vectors as well. The GraphSAGE [25] model has been proposed to turn graph neural network into inductive learning using an aggregation function, which solves the problem of insufficient generalization ability of direct push learning that Deepwalk faces. It also successfully promoted the development of PinSAGE, which is the first industrial graph neural network recommendation system.

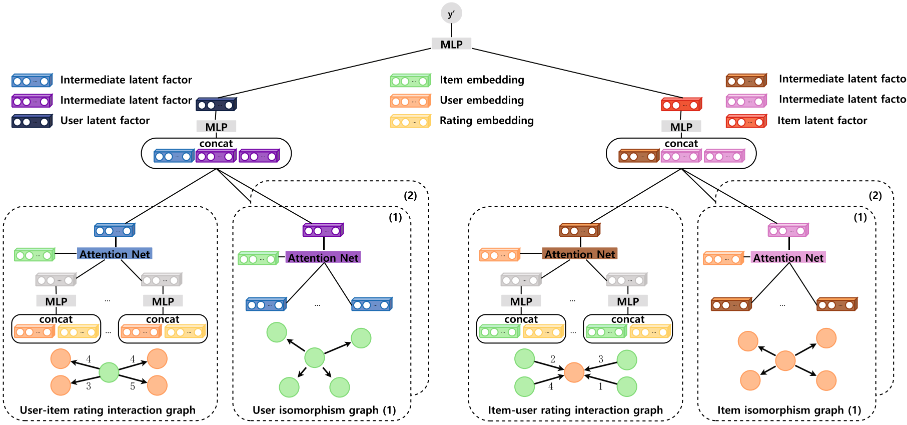

y'
MLP
Intermediate latent factor
Intermediate latent factor
User latent factor
Item embedding
User embedding
Rating embedding
Intermediate latent factor
Intermediate latent factor
Item latent factor
MLP
concat
MLP
concat
(2)
(1)
(2)
(1)
Attention Net
Attention Net
Attention Net
Attention Net
MLP
MLP
concat
concat
MLP
MLP
concat
concat
4
4
3
5
2
3
4
1
User-item rating interaction graph
User isomorphism graph (1)
Item-user rating interaction graph
Item isomorphism graph (1)

Fig. 2 The framework of SGNNRec model

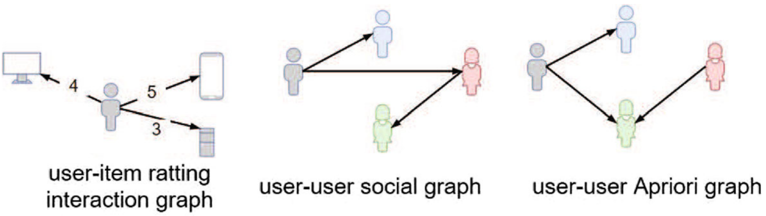


Fig. 3 The graph spaces of user model

3 Proposed Framework

The overall framework of the SGNNRec model is shown in Fig. 2. The SGNNRec consists of three sub-models, which are user model, item model and rating prediction model. The user model is based on the user-item rating graph, supplemented by other user isomorphic graphs to obtain the user latent factors. The user isomorphism graph is obtained by some data mining algorithms that utilize the auxiliary information. Here we construct two isomorphic graphs which are user's social graph and user's interation graph. If we can get more user auxiliary information, we can supplement the information into the SGNNRec model by constructing the isomorphic graphs. Therefore, our model is flexible and scalable. The item model is to obtain the item latent factors, and the method is similar to the user model, which is based on the user-item rating graph and supplemented by the item isomorphism graph. Finally, we input the user latent factor and item latent factor into the rating prediction, and get the user's predicted rating for the item.

3.1 User Model

The main purpose of the user model is to learn the user's latent factor $\mathbf{h}_i$ (the length of $\mathbf{h}_i$ is d). The key issue of user graph construction is how to integrate three different graph spaces (shown in Fig. 3) and fully reflect the influence of different user relationships on the final user interest degree.

The model needs to learn the latent factors under the three graph spaces, namely the latent factor $\mathbf{h}_i^I$ in the item space, the latent factor $\mathbf{h}_i^S$ in the social space, and the latent factor $\mathbf{h}_i^A$ in the apriori space. All their length are d. The methods to obtain these latent factors are as follows.

3.1.1 Item Aggregation

The user-item rating interaction graph is a weighted directed graph. In order to simultaneously consider the interaction and the rating weight between nodes, a solution is proposed.

We aim to obtain a vector that represents the actions as a rating interaction between user u_i and an item i_a denoted as $\mathbf{x}_{ia}$. The user u_i may have many rating interactions; $\{\mathbf{x}_{ia} | \forall a \in C(i)\}$ represents the set of items which user u_i interacted with. To obtain $\mathbf{x}_{ia}$ and consider both the interaction and the rating, we use the following function:

$$\mathbf{x}_{ia} = gv([\mathbf{q}_a \oplus \mathbf{e}_r]) \tag{1}$$

$\mathbf{q}_a$ represents item i_a embedding vector (length d) and $\mathbf{e}_r$ represents rating r embedding vector (length d). $\oplus$ represents the concatenation operator of two vectors. gv is MLP(Multi-Layer Perceptron).

$\mathbf{h}_i^I$ can be obtained by the aggregation function $Aggre_{items}$:

$$\mathbf{h}_i^I = \sigma(W \cdot Aggre_{items}(\{\mathbf{x}_{ia} | \forall a \in C(i)\}) + b) \tag{2}$$

σ is the nonlinear activation function. The common aggregation function in graph neural networks is mean-aggregation or max-aggregation. W and b are the weight and bias of the neural network. Because $\mathbf{x}_{ia}$ represents the user's rating interaction action, and the actions express the user's preferences, different rating interaction actions have different strengths. Therefore, in this study, we use attention-aggregation to incorporate such differences as follows:

$$\mathbf{h}_i^I = \sigma\left(W \cdot \left\{\sum_{a \in C\{i\}} \alpha_{ia} \mathbf{x}_{ia}\right\} + b\right) \tag{3}$$

α_{ia} represents the different strengths of $\mathbf{x}_{ia}$. $\mathbf{x}_{ia}$ represents the user's rating interaction action, which express the user's preferences. We need user u_i embedding vector $\mathbf{p}_i$ (length d). Subsequently, we need to concatenate $\mathbf{p}_i$ and $\mathbf{x}_{ia}$ and develop them into an attention network to learn α_{ia}. The attention network is determined as follows:

$$\alpha_{ia}^* = W_2^T \cdot \sigma\left(W_1 \cdot \left[\mathbf{x}_{ia} \oplus \mathbf{p}_i\right] + b_1\right) + b_2 \tag{4}$$

$$\alpha_{ia} = \frac{\exp\left(\alpha_{ia}^*\right)}{\sum_{a \in C(i)} \exp\left(\alpha_{ia}^*\right)} \tag{5}$$

3.1.2 Social Aggregation

Due to people's preferences to a certain extent depend on their social friends, the preferences of users' friends can be considered when recommending. In graph neural networks, integrating the preferences of friends also needs social aggregation to aggregate the social relationship graph followed by learning the users' latent factors in social space. The details are as follows:

$$\mathbf{h}_i^S = \sigma(W \cdot Aggre_{neighb}(\{\mathbf{h}_o^I | \forall o \in N(i)\}) + b) \tag{6}$$

$\mathbf{h}_o^I$ is the latent factor of a user's friend in the item space. Because a user's friend is also a user, the friends' latent factor in the item space can also be obtained by item aggregation. $N(i)$ is the set of social friends. Through the social aggregation function, all the latent factors of users' friends in the item space are aggregated to obtain the latent factors of users in the social space.

Different friends have different influence on users' preferences, the attention mechanism is embedded in the social aggregation function. The formula is defined as follows.

$$\mathbf{h}_i^S = \sigma\left(W \cdot \left\{\sum_{o \in N\{i\}} \beta_{io}\mathbf{h}_o^I\right\} + b\right) \tag{7}$$

β_{io} represents the strength of user's friend o in the aggregation process. It can also be obtained by the same method used in item aggregation. Because it is $\mathbf{h}_o^I$ strength to user's preference, we also need user u_i embedding vector $\mathbf{p}_i$. Concatenating $\mathbf{p}_i$ and $\mathbf{h}_o^I$, and then integrating them into an attention network, we have β_{io} as follows:

$$\beta_{io}^* = W_2^T \cdot \sigma\left(W_1 \cdot \left[\mathbf{h}_o^I \oplus \mathbf{p}_i\right] + b_1\right) + b_2 \tag{8}$$

$$\beta_{io} = \frac{\exp\left(\beta_{io}^*\right)}{\sum_{o \in C(i)} \exp\left(\beta_{io}^*\right)} \tag{9}$$

3.1.3 Apriori Aggregation

In real life, users often interact with items but do not rate them. For example, in e-commerce shopping, a user purchases many products but rate only a few of them. Thus, a large number of null values appear in the rating matrix, but the interaction between users and items is of great significance to improve the recommendation accuracy. In this study, an apriori aggregation method is proposed to mine the potential similar user group information.

An item that has been interacted with by many users can be regarded as a piece of data $\mathbf{t}_i = \{u_1, u_3, u_5...\}$, and all the items' interacted information can be used as the dataset $\mathbf{S} = \{\mathbf{t}_i | \forall i \in M\}$ for the apriori algorithm. M is the number of items. Through the apriori algorithm, we can determine the frequent itemsets with a length of two. A frequent itemset with length two can be seen as a user pair; we can add a trust relationship between the two users of the user pair. In this way, the user apriori graph is constituted.

Because this graph relationship is similar to a social relationship, we can use the method of social aggregation to aggregate the latent factors $\mathbf{h}_i^A$ in the apriori space.

$$\mathbf{h}_i^A = \sigma\left(W \cdot \left\{\sum_{m \in A\{i\}} \mu_{im}\mathbf{h}_m^I\right\} + b\right) \tag{10}$$

$A(i)$ is the set of users that are neighbors of user u_i in the apriori graph. The attention network to learrn μ_{im} is determined as follows:

$$\mu_{im}^* = W_2^T \cdot \sigma\left(W_1 \cdot \left[\mathbf{h}_m^I \oplus \mathbf{p}_i\right] + b_1\right) + b_2 \tag{11}$$

$$\mu_{im} = \frac{\exp\left(\mu_{im}^*\right)}{\sum_{m \in A(i)} \exp\left(\mu_{im}^*\right)} \tag{12}$$

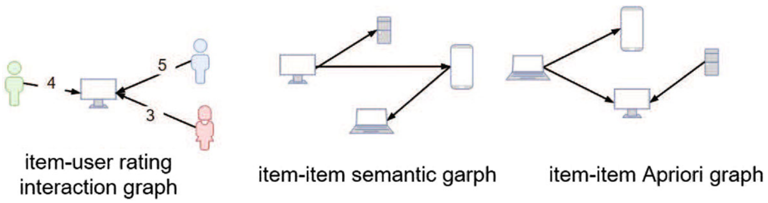


Fig. 4 Item model's three kinds of graph

3.1.4 User Latent Factor

The final latent factors of users are obtained by aggregating the latent factors in the three graph spaces. Considering that different graphs have different strength influence on users' preferences, the latent factors in the three spaces are aggregated by the attention aggregation function, as follows:

$$\delta_{in}^{*} = W_2^T \cdot \sigma\left(W_1 \cdot \left[\mathbf{h}_i^n \oplus \mathbf{p}_i\right] + b_1\right) + b_2 \tag{13}$$

$$\delta_{in} = \frac{\exp\left(\delta_{in}^{*}\right)}{\sum_{n \in \{I,S,A\}} \exp\left(\delta_{in}^{*}\right)} \tag{14}$$

$$\mathbf{h}_i = mlp_{user}\left(\sum_{n \in \{I,S,A\}} \delta_{in} \mathbf{h}_i^n\right) \tag{15}$$

mlp_{user} is a MLP performed on the user latent factor.

3.2 Item Model

The main purpose of the item model is to learn the item latent factor $\mathbf{z}_j$. In the same way, first it learns the three latent factors under the three graph spaces (as shown in Fig. 4). The latent factor $\mathbf{z}_j^U$ in user space, the latent factor $\mathbf{z}_j^M$ in semantic space and the latent factor $\mathbf{z}_j^A$ in apriori space have d length.

3.2.1 User Aggregation

User aggregation is the same as item aggregation when only the item is considered as the subject.

For item i_j, there is a set of users who have rating interacted with it. We denote the set as $B(j)$. First, we need to obtain $\{\mathbf{f}_{jt} | \forall t \in B(j)\}$ represents the rating interaction representation vector of user u_t for item i_j.

The method of obtaining $\mathbf{f}_{jt}$ is similar as the method of obtaining $\mathbf{x}_{ia}$. First, we need to concatenate user u_t embedding vector $\mathbf{p}_t$ and rating r embedding vector $\mathbf{e}_r$ followed by integrating them into MLP gu. Finally, we get $\mathbf{f}_{jt}$as follows:

$$\mathbf{f}_{jt} = gu\left(\left[\mathbf{p}_t \oplus \mathbf{e}_r\right]\right) \tag{16}$$

We obtain $\{\mathbf{f}_{jt} | \forall t \in B(j)\}$, followed by the user aggregation to obtain latent factor $\mathbf{z}_j^U$. It is also the attention aggregation as follows:

$$\mathbf{z}_j^U = \sigma\left(W \cdot \left\{\sum_{t \in B\{j\}} \alpha_{jt}\mathbf{f}_{jt}\right\} + b\right) \tag{17}$$

$$\alpha_{jt}^* = W_2^T \cdot \sigma\left(W_1 \cdot \left[\mathbf{f}_{jt} \oplus \mathbf{q}_j\right] + b_1\right) + b_2 \tag{18}$$

$$\alpha_{jt} = \frac{\exp\left(\alpha_{jt}^*\right)}{\sum_{t \in B(j)} \exp\left(\alpha_{jt}^*\right)} \tag{19}$$

$\mathbf{q}_j$ is the embedding vector of item i_j.

3.2.2 Semantic Aggregation

Considering that people like an item, they would also like other items of a similar kind. The items of the same kind can be considered in the recommendation. Using the tag information of items, we can build the semantic relationship graph of the items. It can help us to find items that are similar to the item that we like. Then, we can add the tag in-formation of items to the model.

Inspired by the work [4], in this study, we first calculate the similarity between tags and obtain the affinity matrix between tags. Second, we cluster the tags by the clustering method. Finally, the clusters of tags are used to calculate the semantic similarity between items. When the semantic similarity is higher than a certain threshold, we add a trust relationship between the two items.

An item has several tags, which we call a resource-term. A tag may exist in multiple resource-terms; the union of these resource-terms is the resource-set of the tag. The similarity calculation of two tags is the intersection and union ratio of resource-sets of two tags. Therefore, the similarity between tags is calculated as follows:

$$sim(t_i, t_j) = \frac{|A_i \cap B_j|}{|A_i \cup B_j|} \tag{20}$$

t_i,t_j represent tag t_i and tag t_j. A_i is the resource-set of tag t_i. B_j is the resource-set of tag t_j. The affinity matrix of tags can be obtained by calculating the similarity between all tags. We could utilize the affinity matrix to cluster tags. Once we obtain tag clusters, we utilize the tag clusters to calculate correlation R_{ik} between item i and tag cluster k. The function is as follows:

$$R_{ik} = \frac{count\ (T_i, C_k)}{size\ (T_i)} \tag{21}$$

$count(T_i, C_k)$ represents the size of sets where the tags of item i intersect the tags of cluster k. $size(T_i)$ represents the number of tags of item i.

After calculating the correlation between all items and all tag clusters, a correlation matrix with the tag clusters as columns and the items as rows can be obtained. This matrix looks like a feature matrix, and each row represents the feature of the item. Then, two items use cosine similarity formula to calculate the semantic similarity between the two items. When the similarity is larger than a certain threshold, we add a trust relationship between the two items. In this way, the semantic graph of items is built.

The semantic graph of items is similar to the social graph of users. It can be regarded as the social graph of the item. Then, the social aggregation in the user model can be used for semantic aggregation of items, as follows:

$$\mathbf{z}_j^M = \sigma\left(W \cdot \left\{\sum_{o \in M\{j\}} \beta_{jo}\mathbf{z}_o^U\right\} + b\right) \tag{22}$$

$M(j)$ is the set of semantic neighbors of item i_j. The attention network to learrn β_{jo} is determined as follows:

$$\beta_{jo}^* = W_2^T \cdot \sigma\left(W_1 \cdot \left[\mathbf{z}_o^U \oplus \mathbf{q}_j\right] + b_1\right) + b_2 \tag{23}$$

$$\beta_{jo} = \frac{\exp\left(\beta_{jo}^*\right)}{\sum_{o \in M(j)} \exp\left(\beta_{jo}^*\right)} \tag{24}$$

3.2.3 Apriori Aggregation

A user that interacts with many items can be regarded as a piece of data $\mathbf{t}_j = \{i_2, i_4, i_6...\}$. Furthermore, all users' interaction information can be used as the dataset $\mathbf{S} = \{\mathbf{t}_j | \forall j \in N\}$ for the apriori algorithm. N is the number of users. Through the apriori algorithm, we can determine the frequent itemsets with a length of two. A frequent itemset with length two can be regarded as an item pair; we can then add a trust relationship between the two items of the item pair. In this way, the item apriori graph is constituted.

The method of aggregation is as follows:

$$\mathbf{z}_j^A = \sigma\left(W \cdot \left\{\sum_{m \in A\{j\}} \mu_{jm}\mathbf{z}_m^U\right\} + b\right) \tag{25}$$

$A(j)$ is the set of items that are neighbors of item i_j in the apriori graph. The attention network to learrn μ_{jm} is determined as follows:

$$\mu_{jm}^* = W_2^T \cdot \sigma\left(W_1 \cdot \left[\mathbf{z}_m^U \oplus \mathbf{q}_j\right] + b_1\right) + b_2 \tag{26}$$

$$\mu_{jm} = \frac{\exp\left(\mu_{jm}^*\right)}{\sum_{m \in A(j)} \exp\left(\mu_{jm}^*\right)} \tag{27}$$

3.2.4 Item Latent Factor

In the same way as the user latent factor, we consider the strength of the three graphs information. We use the attention network to aggregate them as follows:

$$\delta_{jn}^{*} = W_2^T \cdot \sigma\left(W_1 \cdot \left[\mathbf{z}_j^n \oplus \mathbf{q}_j\right] + b_1\right) + b_2 \tag{28}$$

$$\delta_{jn} = \frac{\exp\left(\delta_{jn}^{*}\right)}{\sum_{n\in\{U,M,A\}} \exp\left(\delta_{jn}^{*}\right)} \tag{29}$$

$$\mathbf{z}_j = mlp_{item}\left(\sum_{n\in\{U,M,A\}} \delta_{jn}\mathbf{z}_j^n\right) \tag{30}$$

mlp_{user} is a MLP performed on the item latent factor.

3.3 Rating Prediction

We concatenate $\mathbf{h}_i$ and $\mathbf{z}_j$, and then put them into a MLP. This is followed by linear prediction. We can obtain the prediction rating $r_{ij}^{'}$ that is given by user u_i for item i_j, as follows:

$$g_1 = \left[\mathbf{h}_i \oplus \mathbf{z}_j\right] \tag{31}$$

$$g_2 = \sigma\left(W_2 \cdot g_1 + b_2\right) \tag{32}$$

$$\ldots\ldots$$

$$g_{l-1} = \sigma\left(W_{l-1} \cdot g_{l-2} + b_{l-1}\right) \tag{33}$$

$$r_{ij}^{'} = W_l \cdot g_{l-1} \tag{34}$$

4 Experiment

4.1 Dataset and Metrics

There are few public datasets in the field of recommender systems containing user-item rating information, user social information, item tag information, and user-item interaction information. Therefore, the experimental dataset in our study is crawled from TapTap website. The TapTap dataset includes 2345 users, 12330 items, 46322 item rating information, 8095 social relationships and 5738 items which have tag information. It also has user-item interaction information for 257586 users. For the 46322 item rating information, 80% were used for training and 20% for verification.

The experimental software environment was as follows: Windows 10-64bits, Anaconda 3, python 3.8, and Pytorch 1.6.0. The hardware environment was as follows: CPU: Intel Core (TM) i5-10400f @ 2.90ghz. Memory is 16 GB. GPU: NVIDIA Geforce RTX 2060 6GB.

The metrics used in this paper are MAE (mean absolute error) and RMSE (root mean square error). MAE measures the gap between the predicted rating and the actual rating.

Smaller MAE represents higher accuracy. The formula is as follows:

$$MAE = \frac{\sum_{i=1}^{N} |P_{ui} - R_{ui}|}{N} \tag{35}$$

The RMSE is similar to MAE, which measures the gap between the predicted rating and the actual rating. The formula is as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (P_{ui} - R_{ui})^2} \tag{36}$$

P_{ui} is the prediction rating of user u for item i. R_{ui} is the real rating of user u for item i.

4.2 Experimental Settings

The gv and gu are as follow:

$$gv(x) = W_2^T \cdot \sigma (W_1 \cdot (x) + b_1) + b_2 \tag{37}$$

$$gu(x) = W_2^T \cdot \sigma (W_1 \cdot (x) + b_1) + b_2 \tag{38}$$

The mlp_{user} and mlp_{item} are as follow:

$$mlp_{user}(x) = \sigma \left(W_3 \cdot \left(W_2^T \cdot \sigma (W_1 \cdot (x) + b_1) + b_2\right) + b_3\right) \tag{39}$$

$$mlp_{item}(x) = \sigma \left(W_3 \cdot \left(W_2^T \cdot \sigma (W_1 \cdot (x) + b_1) + b_2\right) + b_3\right) \tag{40}$$

All the σ in this paper are $Relu$.

Because it is a linear prediction of rating, the loss function uses the mean square error:

$$Loss = \frac{1}{2|O|}\sum_{i,j \in O} \left(r'_{ij} - r_{ij}\right)^2 \tag{41}$$

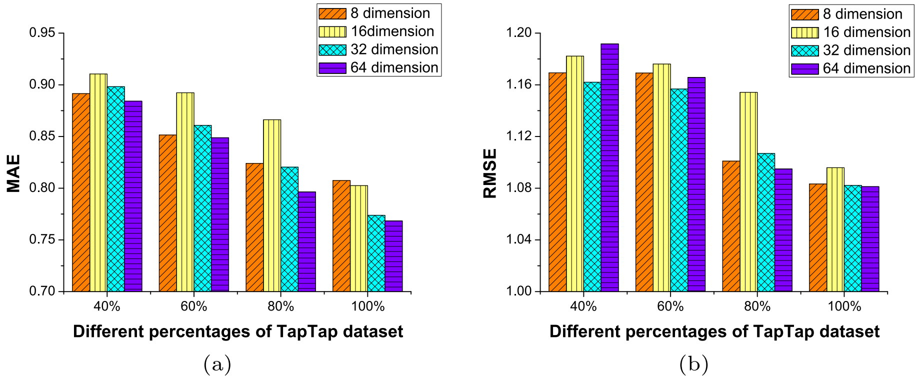
In representation learning, the embedding dimension d is an important parameter. We performed an experiment to choose the best d. The experiment results are as follows (Fig. 5).

We performed this experiment on different percentages of TapTap dataset. At 40%, the data is sparse. Meanwhile, higher dimensions cannot receive sufficient data to learn. As the data increases, the performance of the high-dimensional models becomes better. Hence, 64 dimensions is the optimum selection.

4.3 Model Analysis

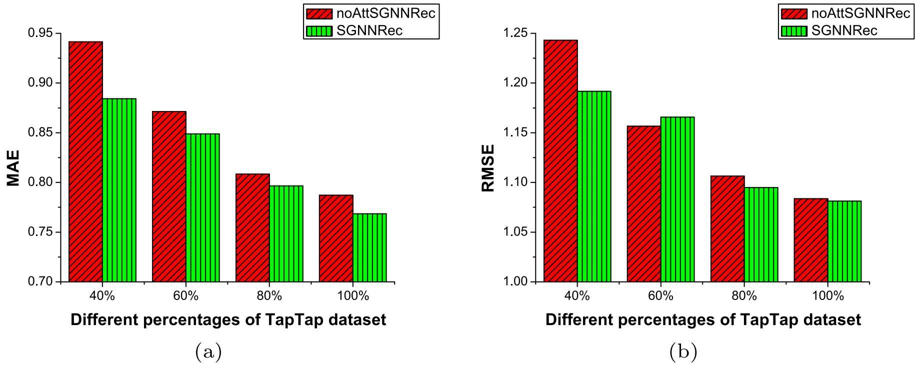
4.3.1 The Effectiveness of the Two-Layer Attention Network

In order to prove the effectiveness of the two-layer attention network structure, we designed the following experiments. The experiment results are as Fig. 6. The "noAttSGNNRec" represents the model structure without the second layer attention mechanism. The experimental results show that models with the second layer attention mechanism can obtain more accurate recommendation results, proving the effectiveness of the double-layer attention network.



(a) (b)

Fig. 5 Recommendation quality of different embedding dimensions. **a** MAE values on different embedding dimensions, **b** RMSE values on different embedding dimensions



(a) (b)

Fig. 6 Recommendation quality of ablation model. **a** MAE values on ablation model, **b** RMSE values on ablation model

4.3.2 Influence of the Different Number of Context Information on the Model

Decomposing our model. SGNNRec0 only utilizes the user-item rating information. Its $\mathbf{h}_i$ and $\mathbf{z}_j$ are as follows:

$$\mathbf{h}_i = \mathbf{h}_i^I \ and \ \mathbf{z}_j = \mathbf{z}_j^U \tag{42}$$

SGNNRec1 utilizes the user-item rating information and user social information. Its $\mathbf{h}_i$ and $\mathbf{z}_j$ are as follows:

$$\mathbf{h}_i = mlp_{user}\left(\sum_{n\in\{I,S\}} \delta_{in}\mathbf{h}_i^n\right) \ and \ \mathbf{z}_j = \mathbf{z}_j^U \tag{43}$$

SGNNRec2 utilizes the user–item rating information, user social information and item tag information. Its $\mathbf{h}_i$ and $\mathbf{z}_j$ are as follows:

$$\mathbf{h}_i = mlp_{user}\left(\sum_{n\in\{I,S\}} \delta_{in}\mathbf{h}_i^n\right) \tag{44}$$

$$\mathbf{z}_j = mlp_{item}\left(\sum_{n\in\{U,M\}} \delta_{jn}\mathbf{z}_j^n\right) \tag{45}$$

SGNNRec3 utilizes the user–item rating information, user social information, item tag information, and user–item interaction information. It is the same as SGNNRec. Its $\mathbf{h}_i$ and $\mathbf{z}_j$ are as follows:

$$\mathbf{h}_i = mlp_{user}\left(\sum_{n\in\{I,S,A\}} \delta_{in}\mathbf{h}_i^n\right) \tag{46}$$

$$\mathbf{z}_j = mlp_{item}\left(\sum_{n\in\{U,M,A\}} \delta_{jn}\mathbf{z}_j^n\right) \tag{47}$$

To verify that the addition of different graph relationship information with different strengths in the model could improve the accuracy and universality of the recommended model, we constructed some models without second layer attention mechanisms as contrast models, which are as follows:

NoattSGNNRec0 only utilizes the user-item rating information. Because there is no context information, it is the same as SGNNRec0. Its $\mathbf{h}_i$ and $\mathbf{z}_j$ are as follows:

$$\mathbf{h}_i = \mathbf{h}_i^I \ and \ \mathbf{z}_j = \mathbf{z}_j^U \tag{48}$$

NoAttSGNNRec1, noAttSGNNRec2 and noAttSGNNRec3 are similar with SGNNRec1, SGNNRec2 and SGNNRec3, while their $\mathbf{h}_i$ and $\mathbf{z}_j$ are different.

NoAttSGNNRec1's $\mathbf{h}_i$ and $\mathbf{z}_j$ are as follows:

$$\mathbf{h}_i = mlp_{user}\left(\mathbf{h}_i^I \oplus \mathbf{h}_i^S\right) \ and \ \ \mathbf{z}_j = \mathbf{z}_j^U \tag{49}$$

NoAttSGNNRec2's $\mathbf{h}_i$ and $\mathbf{z}_j$ are as follows:

$$\mathbf{h}_i = mlp_{user}\left(\mathbf{h}_i^I \oplus \mathbf{h}_i^S\right) \tag{50}$$

$$\mathbf{z}_j = mlp_{item}\left(\mathbf{z}_j^U \oplus \mathbf{z}_j^M\right) \tag{51}$$

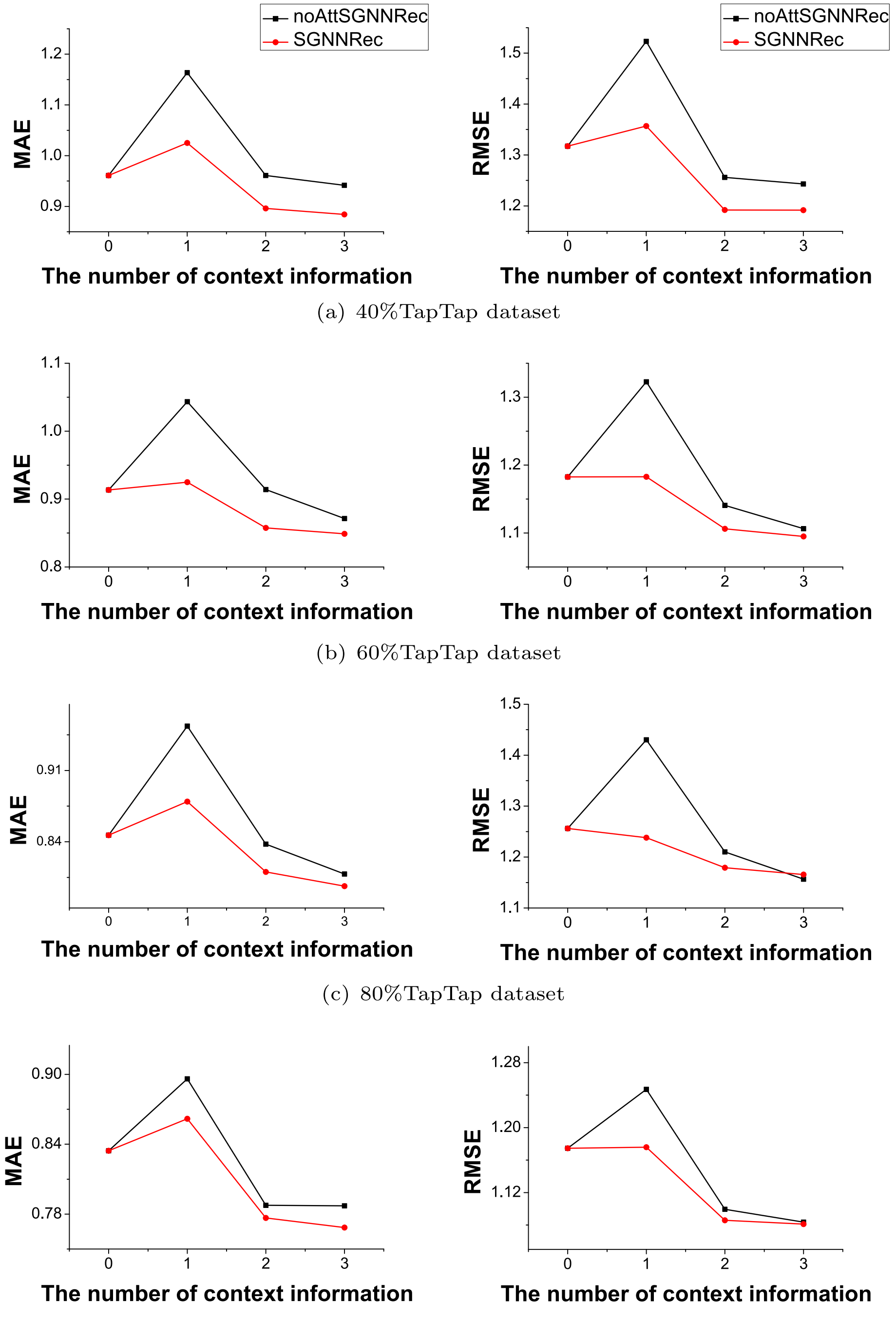
NoAttSGNNRec3's $\mathbf{h}_i$ and $\mathbf{z}_j$ are as follows:

$$\mathbf{h}_i = mlp_{user}\left(\mathbf{h}_i^I \oplus \mathbf{h}_i^S \oplus \mathbf{h}_i^A\right) \tag{52}$$

$$\mathbf{z}_j = mlp_{item}\left(\mathbf{z}_j^U \oplus \mathbf{z}_j^M \oplus \mathbf{z}_j^A\right) \tag{53}$$

NoAttSGNNRec is used as the baseline for comparison. The experimental results are shown in Fig. 7.

The result shows that the quality of recommendation improves when auxiliary information is constantly added. However, the recommendation quality of SGNNRec0 is better than that



(a) 40%TapTap dataset

(b) 60%TapTap dataset

(c) 80%TapTap dataset

(d) 100%TapTap dataset

Fig. 7 SGNNRec's and noAttSGNNRec's results on different number of context information. **a** the dataset is 40% TapTap dataset. **b** The dataset is 60% TapTap dataset. **c** the dataset is 80% TapTap dataset. **d** the dataset is 100% TapTap dataset

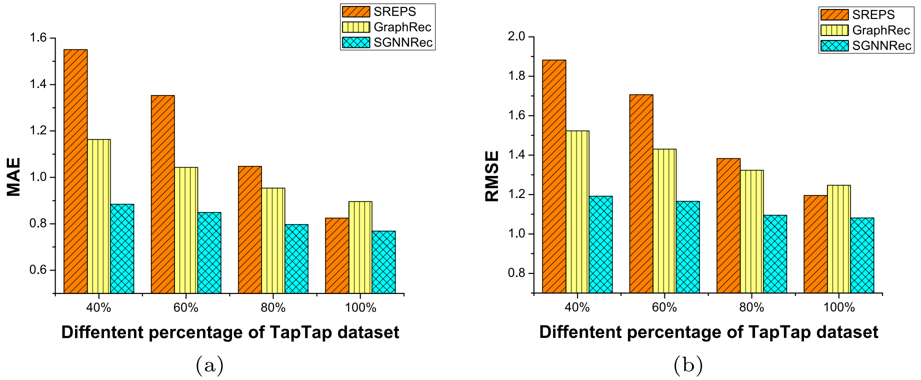


Fig. 8 Performance of three models on different percentages of TapTap dataset

of SGNNRec1 and noattSGNNRec1, which indicates that the recommendation quality of the model would be lower when poor quality social information is added. The quality of SGNNRec1 is better than that of noattSGNNRec1, which indicates that considering the different strengths of different auxiliary information can reduce the influence of poor quality auxiliary information on the model and improve the recommendation quality.

With the addition of social information, the error of the model generally increases. This shows that the quality of social information is low, which will negatively affect the recommendation results. But the error improvement of NoAttSGNNRec is more than that of SGNNRec. It shows that low-quality social information has a greater negative impact on NoAttSGNNRec that does not consider the different quality of different graph information. However, considering that different graph information has different quality, SGNNRec has little or no increase in error when social information is added. It prove that two-layer attention network can enhance the universality of our model.

With the addition of label information and user-item interaction information, the error of recommendation results becomes lower and lower. This proves that our proposed methods of constructing an item isomorphic graph with item labels and constructing isomorphic graphs of users and items with user-item interaction information to are effective.

4.3.3 The Effectiveness of Sparse Data Recommendation

We performed the experiment on different percentages of TapTap dataset. We mainly compared three models. They are SREPS [26], GraphRec [9], and SGNNRec. Figure 8 shows the result of the comparison.

First, this experiment proves that our model is better than the GNN model GraphRec, and is also better than the deep learning model SREPS. Second, it proves that the auxiliary information is effective. When data is sparse, our model is far better than the two other models because it has sufficient auxiliary information as a supplement. This proves that our model can solve the problems of data sparsity and cold start problems well.

4.4 Performance Comparison of Recommendation Systems

Five algorithms are selected for comparison:

Table 1 Dataset information

	TapTap	Douban
Users	2345	5416
Items	12330	15000
Rating	46322	198751
Interaction	257586	254386
Items with tags	5738	14650
Tags	811	15733
Social Link	8095	4421
Density	0.1602%	0.2446%

Table 2 Models comparison

	TapTap		Douban	
Algorithms	MAE	RMSE	MAE	RMSE
NMF	0.9017	1.1932	0.6320	0.8276
PMF	0.9127	1.1901	0.6430	0.8451
SVD	0.8362	1.1218	0.6079	0.7895
SREPS	0.8245	1.1950	0.6271	0.8378
GraphRec	0.8961	1.2472	0.6000	0.7920
ConsisRec	0.9076	1.1523	0.6124	0.7906
SGNNRec	**0.7684**	**1.0812**	**0.5970**	**0.7844**

NMF: uses Non-negative Matrix Factorization to factor the user-item rating matrix so as to rate non-rated items.

PMF [27]: Probabilistic Matrix Factorization utilizes user-item rating matrix only and models latent factors of users and items by Gaussian distributions.

SVD: SVD is based on Singular Value Decomposition. It obtains hidden characteristics of users and items as feature vectors and utilizes the feature vectors to predict rating.

SREPS [26]: This is an embedded learning method based on social network that was proposed in the 2018 AAAI conference to represent and learn social networks to make recommendations.

GraphRec [9]: In the 2019 WWW conference, a neural network recommendation system was proposed, which utilizes the user-item rating graph and user social graph.

ConsisRec [28]: ConsisRec is a neural network recommendation model which uses neighbor sampling to solve the social inconsistency problem. It was proposed in the 2021 SIGIR conference.

To verify the universality of the model, we performed experiments on two datasets. Table 1 shows the information of the two datasets.

Table 2 shows the MAE and RMSE for each model on the TapTap dataset and Douban dataset. The experimental results show that our model is better than the traditional matrix decomposition method, deep learning recommendation model, and graph neural network recommendation model. The bold section in the table shows the experimental results of our SGNNRec model. The smaller the value represented in bold, the better the recommendation performance of the model. It can be seen that the MAEMAE (mean absolute error) and RSME (root mean square error) of the SGNNRec model are both smaller than other models, indicating that the recommendation accuracy of the model proposed in this paper is higher.

Because the quality of TapTap dataset's social information is poor, GraphRec only uses social relationships as a supplement, which does not consider the different strengths of different information. Hence, its recommendation results are not as good as traditional algorithms when the dataset has poor social information (This was proved in Sect. 4.3). However, the model in this paper uses multiple auxiliary information and achieves good results, which shows that a large amount of auxiliary information can improve the quality of recommendation.

Owing to the density of Douban dataset being higher than that of TapTap dataset, the GNN model GraphRec can obtain sufficient neighbor nodes to learn. Its performance is better than that of the traditional matrix decomposition method. However, Our model adds more auxiliary information and considers more comprehensive information, resulting in more comprehensive results. Therefore our model achieves the best accuracy.

5 Conclusion

This paper proposes a model, which can make use of different auxiliary information in an easy and quick way. Experimental results show our model can obtain high-quality recommendation results though the rating information is sparse. And our model adds some auxiliary information as a supplement, which improves the quality of recommendation. It can reduce the impact of low-quality auxiliary informational information on the quality of recommendation. In addition, we proposed methods to convert item tag information and user-item interaction information into isomorphic graphs. In the future, we can construct other isomorphic graphs with additional auxiliary information such as image information (items poster) and text information (user comment). Adding more auxiliary information as supplements for recommendation can result in a more comprehensive and high-quality recommendation system.

Acknowledgements The authors would like to thank the editor and the anonymous reviewers for their constructive and valuable comments. This work was supported in part by the National Natural Science Foundation of China under Grant No. 62162067, No. 82360280, the Open Foundation of Key Laboratory in Software Engineering of Yunnan Province under Grant No. 2020SE310, the Open Foundation of Engineering Research Center of Cyberspace under Grant No. KJAQ202112013, and the Science and Technology Plan in Key Fields of Yunnan under Grant 202001BB050076.

References

1. Edmunds A, Morris A (2000) The problem of information overload in business organisations: a review of the literature. Int J Inf Manag 20(1):17–28
2. Cai Q, Han D, Li H, Hu Y, Chen Y (2014) Personalized resource recommendation based on tags and collaborative filtering. Comput Sci 41(1):69–71

3. Hui S, Pengyu L, Kai Z (2011) Improving item-based collaborative filtering recommendation system with tag. In: 2011 2nd international conference on artificial intelligence, management science and electronic commerce (AIMSEC). IEEE, pp 2142–2145
4. Li H, Fu Y (2018) Collaborative filtering recommendation algorithm based on tag clustering and item topic. Comput Sci 45(4):247–251
5. Jin J, Qin J, Fang Y, Du K, Zhang W, Yu Y, Zhang Z, Smola AJ (2020) An efficient neighborhood-based interaction model for recommendation on heterogeneous graph. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, pp 75–84
6. Hong H, Lin Y, Yang X, Li Z, Fu K, Wang Z, Qie X, Ye J (2020) Heteta: heterogeneous information network embedding for estimating time of arrival. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, pp 2444–2454
7. Xu J, Zhu Z, Zhao J, Liu X, Shan M, Guo J (2020) Gemini: a novel and universal heterogeneous graph information fusing framework for online recommendations. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, pp 3356–3365
8. Berg Rvd, Kipf TN, Welling M (2017) Graph convolutional matrix completion. arXiv preprint arXiv:1706.02263
9. Fan W, Ma Y, Li Q, He Y, Zhao E, Tang J, Yin D (2019) Graph neural networks for social recommendation. In: The world wide web conference, pp 417–426
10. Wu Q, Zhang H, Gao X, He P, Weng P, Gao H, Chen G (2019) dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. In: The world wide web conference, pp 2091–2102
11. Wang M, Lin Y, Lin G, Yang K, Wu X-M (2020) M2grl: a multi-task multi-view graph representation learning framework for web-scale recommender systems. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, pp 2349–2358
12. Hu B, Zhou N, Zhou Q, Wang X, Liu W (2020) Diffnet: a learning to compare deep network for product recognition. IEEE Access 8:19336–19344
13. Zhang M, Chen Y (2020) Inductive matrix completion based on graph neural networks. In: International conference on learning representations
14. Wu L, Li J, Sun P, Hong R, Ge Y, Wang M (2020) Diffnet++: a neural influence and interest diffusion network for social recommendation. IEEE Trans Knowl Data Eng
15. Wei Y, Ma H, Zhang R, Li Z, Chang L (2021) Exploring implicit relationships in social network for recommendation systems. In: Proceedings of the 25th Pacific-asia conference on knowledge discovery and data mining (PAKDD), pp 386–397
16. Mandal S, Maiti A (2022) Heterogeneous trust-based social recommendation via reliable and informative motif-based attention. In: Proceedings of the 2022 international joint conference on neural networks (IJCNN), pp 1–8
17. Yu J, Yin H, Li J, Gao M, Huang Z, Cui L (2022) Enhance social recommendation with adversarial graph convolutional networks. IEEE Trans Knowl Data Eng
18. Mandal S, Maiti A (2021) Graph neural networks for heterogeneous trust based social recommendation. In: Proceedings of the 2021 international joint conference on neural networks (IJCNN), pp 1–8
19. Kanungo T, Mount DM, Netanyahu NS, Piatko CD, Silverman R, Wu AY (2002) An efficient k-means clustering algorithm: analysis and implementation. IEEE Trans Pattern Anal Mach Intell 24(7):881–892
20. Ester M, Kriegel H-P, Sander J, Xu X, et al (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Kdd, vol 96, pp 226–231
21. Von Luxburg U (2007) A tutorial on spectral clustering. Stat Comput 17(4):395–416
22. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781
23. 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
24. Barkan O, Koenigstein N (2016) Item2vec: neural item embedding for collaborative filtering. In: 2016 IEEE 26th international workshop on machine learning for signal processing (MLSP). IEEE, pp 1–6
25. Hamilton WL, Ying R, Leskovec J (2017) Inductive representation learning on large graphs. arXiv preprint arXiv:1706.02216
26. Liu C-Y, Zhou C, Wu J, Hu Y, Guo L (2018) Social recommendation with an essential preference space. In: Proceedings of the AAAI conference on artificial intelligence, vol 32
27. Mnih A, Salakhutdinov RR (2007) Probabilistic matrix factorization. Adv Neural Inf Process Syst 20:1257–1264

28. Yang L, Liu Z, Dou Y, Ma J, Yu PS (2021) Consisrec: enhancing GNN for social recommendation via consistent neighbor aggregation. In: Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval, pp 2141–2145