International Journal of Computer Science and Information Technology

ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 3, Number 1, Year 2024
DOI: https://doi.org/10.62051/ijcsit.v3n1.21
Journal homepage: https://wepub.org/index.php/IJCSIT/index

Similarity-based Graph Convolution Collaborative Recommendation Approach

Wentao Zhao, Dewang Wang *

School of Computer Science and Technology, Henan Polytechnic University, Jioazuo, Henan 454000, China

ABSTRACT

With the rapid and iterative development of science and technology, a large amount of information exceeds the range that can be accepted, processed, or effectively utilized by an individual or a system, and recommendation algorithms can, to a certain extent, solve such problems, but traditional recommendation algorithms do not have a good solution to the problems related to data sparsity and recommendation accuracy. A similarity-based graph convolutional neural collaborative recommendation method (GCSCF) is proposed. The similarity algorithm based on the attributes of the item is used to find the item with the highest similarity and has not interacted with the current user, and this item is set to interact with the current user. The relevant interaction information of the user and the item is converted into relative feature vectors; the feature vectors are propagated using a graph convolutional neural network to aggregate the localized information, and the weight coefficients based on the item ratings are normalized to reduce the noise caused by the information aggregation. Comparative experiments are conducted on two public datasets, MovieLens-1M and Movielens-100K, with five baseline models on the set, and using Recall, Normalized Discounted Cumulative Gain (NDCG), and Precision as the evaluation metrics, and the results of the experiments show that the performance of the proposed social recommendation model better than other models.

KEYWORDS

Recommender systems; Deep learning; Collaborative recommendation; Item attributes; Graph convolutional neural networks

1. INTRODUCTION

With the rapid popularization of the Internet in people's daily life, data and information have ushered in explosive growth, resulting in users' interests and intentions being blurred, making it difficult to find the effective information they need, thus generating problems such as information overload [1]. How to help users quickly filter out effective and accurate information from massive data has become one of the urgent problems for researchers to solve. Recommender systems then came into being to effectively alleviate information overload and satisfy users' personalized recommendations [2].

Collaborative filtering (CF) [3], the most commonly used computation in recommender systems, is in essence similar to: “Things are grouped together, people are grouped together”, which essentially utilizes the interaction information between users and items, and algorithmically searches the dataset for similar users and projects in the dataset through algorithms. However, there are disadvantages such as difficulty in adapting to sparse data, cold start and poor scalability [4]. In order to effectively solve the above problems, researchers have classified the current recommendation algorithms into three categories based on their studies of recommender systems: user-based recommendation algorithms [5], item-based recommendation algorithms [6], and hybrid recommendation [7-8].

Since hybrid recommendation algorithms are traditional recommendation algorithms fused with various machine learning for recommendation, they can effectively deal with the shortcomings of traditional algorithms such as adapting to sparse data, cold-start and poor scalability [9-10]. However, since traditional machine learning can only deal with conventional data structures, it is difficult to adapt to the increasing number of Euclidean structured data (Euclidean structured data is characterized by a fixed regularity in the order of arrangement and a corresponding regularity in the frequency of occurrence) that appear on the Internet [11], so it adopts graph convolutional neural networks, which use Laplace operators to convert Euclidean structured data from the time domain to the frequency domain. zhang et al. et al [12] proved through a large number of experiments that the hidden higher-order information in Euclidean structured data can be mined by multiple graph convolutional neural networks.

Literature [12] proposes the GCN algorithm to characterize the interactions between real features in the real world based on the spatial dependencies of the graph structure, and generates new node feature representations by aggregating the node features, which is a scalable semi-supervised learning approach that enables the GCN to perform convolutional iterations directly on the graph structure. Literature [13] proposes a deep learning model based on heterogeneous graph convolutional networks, called HGCN, for classifying entities in (Heterogeneous Information Network) HINs. The model mines potential information from complex HINs through multi-layer heterogeneous convolution and can directly process the original HINs without converting the network from heterogeneous to isomorphic, thus maintaining fine-grained relational semantics of different types of nodes. Literature [14] proposes a graph neural collaborative filtering algorithm, NGCF, which improves the performance and accuracy of the recommender system by aggregating the neighborhood feature information in an iterative manner, so that the higher-order hidden information in the user-item interaction graphs can be effectively mined and utilized.

Due to the above traditional collaborative filtering recommendation algorithms exist poor recommendation accuracy and cold start as well as graph convolutional neural networks face the problems of parameter redundancy, long training time, and failure to mine the higher - order hidden relationships between similar users or similar items. In this paper, we propose a hybrid recommendation model (Similarity - based Graph Convolutional Neural Collaborative Recommendation Method, GCSCF) that combines item attribute-based collaborative filtering algorithm and graph convolutional network to capture the higher-order relationships between similar users or similar items by utilizing collaborative filtering algorithm. Higher - order relationships between similar items, iterating the user-item interaction relationship information through multiple layers to come up with feature representations about the relationship between users and items, and at the same time normalizing and redistributing the weights. The main contributions of this paper are as follows:

Combining the traditional collaborative filtering recommendation algorithm with graph convolutional networks, a new model, GCSCF, is proposed to enhance the accuracy of the recommendation system.

Effectively solve the problem that traditional collaborative filtering recommendation algorithms are difficult to adapt to sparse datasets and efficiently deal with graph convolutional neural networks are difficult to deal with the hidden information of similar users and items, and improve the higher-order relationship between users and items.

The validity of the present model was verified on two real datasets, Movielens-1M and Movielens-100K.

2. RELATED WORK

2.1. Traditional Collaborative Filtering Recommendation Algorithms

The essence of traditional collaborative filtering recommendation algorithms is to perform similarity calculations on all users or items in the data set, and recommend similar items or users to users by ranking. It can be broadly categorized into three types, which are user-based collaborative filtering recommendation algorithms, item-based collaborative filtering recommendation algorithms, and matrix decomposition-based collaborative filtering recommendation algorithms [15-17].

User-based collaborative filtering recommendation algorithm is scored by the interaction record between the user and the item, and its essence when the user needs personalized recommendation, it can first find the cluster of users with similar interests to him, and then the items that the cluster of users with similar interests like and that the user does not have are recommended to the user, and this is the user-based collaborative filtering [18-19].

Item-based collaborative filtering recommendation algorithm is scored by the correlation information between the items, the essence of which is to perform similarity calculation through various labels between items or interaction records between items and users, and then recommend the similar items to the users [20-21].

Collaborative filtering recommendation algorithm based on matrix decomposition can effectively solve the shortcomings of item-based or user-based collaborative filtering recommendation algorithms to mine the effective information hidden in the data is too weak, the essence of which is to decompose the user-item interaction graph into a number of matrices multiplied together, and the decomposed feature matrix is used as the user's feature matrix, and the recommendation is carried out in accordance with the feature matrix [22-23].

2.2. Collaborative Filtering Recommendation Algorithm Based on Graph Neural Networks

In order to more effectively address the reduction in recommendation accuracy due to sparse data, researchers have thus created a neural network capable of processing graph data through the idea of convolutional neural networks, named Graph Neural Network (GNN) [24].

Tang et al [25] proposed the LINE recommendation algorithm in order to solve the embedding of large information networks into low-dimensional vector spaces, which introduces graph neural networks into recommender systems and the algorithm can be applied to any kind of graphs. Leskovec et al [26] proposed the PinSAGE algorithm model, which represents the feature vectors of the nodes by combining the randomized wandering with graph convolutional neural networks and takes into account both the graph structure and the node feature information. Feature vectors, taking into account both graph structure and node feature information. The graph convolutional neural network is successfully applied to industrial-grade recommender systems. Tang et al [27] proposed the NNCF algorithm model, which is based on the combination of neighborhood-based collaborative filtering and neural system, integrating the hidden information in the neighborhood into the neural collaborative filtering, including the history of interaction between the item and the user, so as to make recommendations. Wang et al [28] proposed the NGCF recommendation algorithm, which can be used for recommending items and users by defining the graph convolutional neural network in the null domain. Defining the graph convolution in the null domain, taking the feature information generated from the interaction information between the user and the item through multiple iterations and defining it as higher-order collaborative information, and performing cross-involution of the resulting higher-order features for recommendation. LightGCN [29] recommendation algorithm is a simplification of the NGCF recommendation algorithm in terms of the information transfer rules, removing the learnable parameters in the model that have a negative impact on the results and the nonlinear transformations.

2.3. Summarize

To summarize, traditional collaborative filtering algorithms have the characteristics of fast deployment, on-line and training. However, traditional collaborative filtering algorithms have disadvantages such as their poor accuracy and difficulty in adapting on sparse matrices. Graph convolutional neural network can effectively improve recommendation accuracy and enhance scalability. And it can effectively deal with the hidden information existing in the sparse matrix.

3. GCSCF MODEL

3.1. Overall Framework of the GCSCF Model

The GCACF model adopts the similarity calculation method based on the project attributes, the higher-order hidden information in the similar projects will be initially aggregated, and then the initial user project history interaction information will be one-hot encoded, and the one-hot encoding will be compressed into dense vectors, and the graph convolutional network will be used to make the dense vectors iterated many times with the output of the feature information, which will be normalized to the weight redistributed to reduce the noise brought about by the aggregation of the information, and then finally, the final feature representation will be cross-interpolated to carry out the recommendation, and the overall architecture of the GCSCF model is shown in Fig. 1.

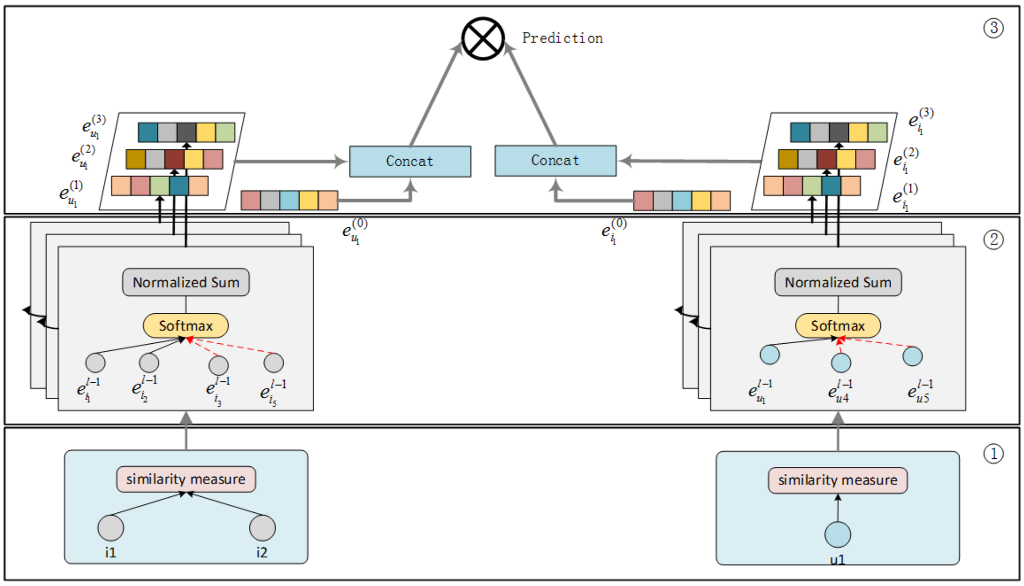


Figure 1. GCSCF modeling framework

The GCSCF model runs through three modules to produce recommendation results, module ① is the similar items aggregation phase; module ② is the graph convolutional neural network that trains the initial vectors for several iterations; and module ③ is the model prediction phase. Among them, module ① is to calculate the maximum similarity items of the items interacted by each user through the similarity function and treat them as neighbor nodes whose weights are proportional to the similarity. At the same time, this step of aggregation operation adds new nodes, so a small amount of noise information will be generated, so module ② uses weight normalization and redistribution to remove noise information.

3.2. The GCACF Model

3.2.1. Similarity calculation

Taking the Movielens-1M dataset as an example, let the set of users be U={$u_1,u_2,...,u_m$}, and the set of movies is I={$i_1,i_2,...,i_n$}, where n and m are the number of users u and movies i, respectively, and the user-movie interaction matrix is R={ $r_{ui} | u \in U, i \in I$ }. where r_{ui} is the rating interaction record of user and movie, the r_{ui} value can be in the range of 0 to 5,if user u has not rated item i, then $r_{ui} = 0$.

Movie tags contain user's interests and preferences, and user's interests are obtained by analyzing the characteristics of movie tags. Among the movie tags can be roughly categorized as action, adventure, animation, comedy, crime, record, drama, science fiction, suspense, etc. Assuming that movies have n attributes, the item attribute matrix is shown in Table 1.

Table 1. Matrix of project attributes

Items	Attributes			
	A_1	A_2	…	A_n
I_1	1	1	…	1
I_2	1	0	…	0
…	…	…	…	1
I_m	0	1	…	0

In this table, {I_1, I_2, ..., I_m} denotes the set of items and {A_1, A_2, ..., A_n} are the attributes of the items. If an item I_i has an attribute A_m, the corresponding item attribute value is 1, otherwise the value is 0. Therefore, in this paper, the similarity based on item attributes is obtained by the following formula:

$$sim_a(\mathrm{I}_i, \mathrm{I}_j) = \frac{N_a^{I_i \cap I_j}}{N_a - N_a^{\overline{I_i \cup I_j}}} \tag{1}$$

where $N_a^{I_i \cap I_j}$ denotes the number of common attributes between I_i and I_j, N_a denotes all the attributes included in the set, and $N_a - N_a^{\overline{I_i \cup I_j}}$ denotes the total set of attributes that belong to neither I_i nor I_j.

Pearson's correlation is a measure of the degree of linear correlation between two variables. It is a value between 1 and -1, where 1 means that the variables are completely positively correlated, 0 means irrelevant, and -1 means completely negatively correlated. In collaborative filtering algorithms, Pearson's correlation can be used to calculate the size of the correlation between two users or two items; the higher the correlation coefficient, the greater the similarity between the two, and vice versa, the smaller the similarity, The formula is as follows:

$$sim_{pearson}(\mathrm{I}_i, \mathrm{I}_j) = \frac{\sum_{i \in I_{uv}} (\mathrm{r}_{ui} - \overline{\mathrm{r}}_i)(\mathrm{r}_{uj} - \overline{\mathrm{r}}_j)}{\sqrt{\sum_{i \in uv} (\mathrm{r}_{ui} - \overline{\mathrm{r}}_i)^2} \sqrt{\sum_{i \in uv} (\mathrm{r}_{uj} - \overline{\mathrm{r}}_j)^2}} \tag{2}$$

Where r_{ui} is the rating of user u for item i and $\overline{r}_{ui}$ is the average rating of user u for item i. I_{uv} denotes the set of items for which users u and v have a common rating.

The Pearson similarity algorithm and the attribute-based similarity algorithm are combined by means of a weighted control coefficient approach, which reduces the influence of movie ratings and movie attributes on the final similarity calculation. The formula for weighted control factor is as follows:

$$\mu_a = \frac{sim_a(\mathrm{I}_i, \mathrm{I}_j)}{sim_a(\mathrm{I}_i, \mathrm{I}_j) + sim_{pearson}(\mathrm{I}_i, \mathrm{I}_j)} \tag{3}$$

$$\mu_p = \frac{sim_{pearson}(\mathrm{I}_i, \mathrm{I}_j)}{sim_a(\mathrm{I}_i, \mathrm{I}_j) + sim_{pearson}(\mathrm{I}_i, \mathrm{I}_j)} \tag{4}$$

Where $\mu_a, \mu_p \in [0,1]$, $\mu_a + \mu_p = 1$. After calculating the weighted control coefficients. Combining the above formulas, the combined project similarity is given by the following equation:

$$sim'(\mathrm{I}_i, \mathrm{I}_j) = \mu_p sim_{pearson}(\mathrm{I}_i, \mathrm{I}_j) + \mu_a sim_a(\mathrm{I}_i, \mathrm{I}_j) \tag{5}$$

In Eq. When all the users in the dataset have not rated the movie, the similarity between the movies can be calculated by the movie attribute, i.e., μ_p is set to 0. This can effectively solve the cold-start problem of the recommender system.

The item that has not interacted with this user and has the highest similarity is embedded in the user item rating matrix this rating is the value of the user's rating of the target item multiplied by the similarity. The formula is as follows:

$$r_{u_i j} = r_{u_i i} sim'(\mathrm{I}_i, \mathrm{I}_j) \tag{6}$$

Where $r_{u_i j}, r_{u_i i} \in [1,5]$ is the rating of user Ui for item I_i and item I_j, respectively, and $sim'(\mathrm{I}_i, \mathrm{I}_j)$ is the similarity value of item I_i and item I_j.

3.2.2. Graph Convolution Operations

The graph convolution operation is used to mine the existence of higher-order multidimensional information in the history of users and items, while the weight coefficients based on item scores are normalized to reduce the noise caused by information aggregation. The propagation rule is: the interaction matrix R is one-hot coded for feature representation, and then the feature representation is embedded according to a certain dimension to get the initial feature vector $e_u^{(0)}$ of the user and the initial feature vector $e_i^{(0)}$ of the item, and then the embedding is propagated using the user-item interaction graph as:

$$e_u^{(k+1)} = \sum_{i \in N_u} \frac{\rho(\mathrm{r}_{ui})}{\sqrt{|N_u|}\sqrt{|N_i|}} e_i^{(k)} \tag{7}$$

$$e_i^{(k+1)} = \sum_{u \in N_i} \frac{\rho(r_{ui})}{\sqrt{|N_i|}\sqrt{|N_u|}} e_u^{(k)} \tag{8}$$

Where $e_u^{(k)}$ and $e_i^{(k)}$ denote the embedding of user u and item i after propagation in layer k, respectively, N_u denotes the set of items that user u interacts with, N_i denotes the set of users that interact with item i, and $\rho(r_{ui})$ is the representation of the item scores that are softmax-normalized, computed as follows:

$$\rho(r_{ui}) = \frac{\exp(r_{ui})}{\sum_{j \in N_u} \exp(r_{uj})} \tag{9}$$

3.2.3. Algorithmic prediction (layer combination and model prediction)

The initial feature vector $e_u^{(0)}$ of the user and the initial feature vector $e_i^{(0)}$ of the item are taken as the unique trainable parameters. The feature representations of each layer reflect the different preference interests of users, so the feature representations of users and items generated in all layers are combined into the final feature representation by linking them through weighted summation to form the final representation of users (items) as follows:

$$e_u = \sum_{k=0}^{K} \alpha_k e_u^{(k)} \tag{10}$$

$$e_i = \sum_{k=0}^{K} \alpha_k e_i^{(k)} \tag{11}$$

Where $\alpha_k \geq 0$ denotes the weight of the kth layer embedding in the composition of the final embedding. In experiments, it is found that setting α_k uniformly to $\frac{1}{K+1}$ usually lead to good performance and greatly simplifies the model complexity. Then the final feature vectors of the user and item are subjected to inner product operation to obtain the user's preference for the movie $\hat{y}$, and the formula is defined as follows:

$$\hat{y}_{ui} = e_u^T e_i \tag{12}$$

3.2.4. Loss function

Since the model uses a collaborative filtering algorithm framework based on matrix decomposition and the loss function of the BPR recommendation algorithm considers the predicted value of positive samples to be greater than the predicted value of negative samples, the loss function in the BPR recommendation algorithm is used to optimize the parameters in the model [42].The loss function of the GCSCF model is:

$$L = \sum_{(u,i,j)\in Z} -\ln\sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \varepsilon\|\upsilon\|_2^2 \tag{13}$$

Where $Z = \{(u,i,j) \mid (u,i) \in Z^+, (u,j) \in Z^-\}$ denotes all the data in the dataset, Z^+ is the positive sample data in the dataset, and Z^- is the negative sample data in the dataset; υ is all the trainable parameters in the GCSCF model; ε is the hyper-parameter determined by the customized cross-validation method used for the model, and penalizing the parameters in the model by ε can effectively avoid model overfitting.

4. EXPERIMENTATION AND ANALYSIS

4.1. Data Set

In this section, the evaluation of this model is carried out by conducting experiments using two public movie datasets Movielens-1M and Movielens-100k with different interaction information, where the dataset contains ratings as well as various labels, and by carrying out experiments with the two interaction information.

Movielens-1M (https: //grouplens.org/datasets/movielens) is a dataset for the scenario of movie recommendation for a larger group of users, which contains about 836,478 interaction records on the MovieLens website consisting of 6,040 users and 3,952 movies, with a score range of The score ranges from 1 to 5, and the sparsity is 95.8%. Using 3-kernel filtering in Movielens-1M, the final dataset contains 6,040 users, 3,629 movies and 836,478 interactions.

Movielens-100K (https://grouplens.org/datasets/movielens) is a dataset for the movie recommendation scenario with a large number of movies, which contains about 100836 interactive records on the MovieLens website consisting of 609 users and 9741 movies, with a score range of 1~5 and a sparsity of 93.7%. The score range is from 1 to 5, and the sparsity is 93.7%. The same 3-kernel filtering method is used in Movielens-100K, and the final dataset contains 609 users, 9741 movies and 100836 interactions.

Table 2. The statistics of the Datasets

Data set	User	Item	Record	sparsity
Movielens-1M	6040	3952	836478	95.8%
Movielens-100K	609	9741	100836	93.7%

4.2. Evaluation Indicators

In order to evaluate the performance of recommendation algorithms by using three classical metrics for recommendation task evaluation: the Normalized Discounted Cumulative Gain (NDCG), the Recall, and the Precision, which are formulated as follows:

The normalized discounted cumulative gain is designed to allow the more highly ranked results to influence the final result:

$$NDCG = \frac{DCG}{IDCG} \tag{14}$$

$$DCG = \sum_{i=1}^{n} \frac{2^{rel_i} - 1}{\log_2(i+1)} \tag{15}$$

$$IDCG = \sum_{i=1}^{|REL|} \frac{2^{rel_i} - 1}{\log_2(i+1)} \tag{16}$$

Where rel_i represents the relevance at the position i; $|REL|$ means that the results are sorted in descending order of relevance, taking the set consisting of the top p results.

Recall is a method of calculating the fraction of relevant items in a dataset:

$$\mathrm{Re}\,call = \frac{TP}{TP + FN} \tag{17}$$

Accuracy can reflect the percentage of recommended items generated by the current recommendation algorithm that are of interest to the user:

$$\mathrm{Pr}\,ecision = \frac{TP}{TP + FP} \tag{18}$$

Where TP (True Position) indicates that positive samples are predicted to be positive; TN (True Negative) indicates that negative samples are predicted to be negative; FN (False Negative) indicates that positive samples are predicted to be negative; and FP (False Positive) indicates that negative samples are predicted to be positive.

4.3. Baseline Model

The validity of the GCSCF model is verified by comparing it with the following five representative benchmark models.

IRACF [30]: Proposes a collaborative filtering algorithm model based on item scores that combines item similarity with item scores through weighting coefficients.

MF [31]: proposes a Bayesian personalized ranking criterion, which is different from the traditional one based on the user rating matrix in that it adopts the implicit feedback from users (e.g., clicks, favorites, etc.), and generates recommendations through the final ranking obtained from the Bayesian analysis of the problem.

NNCF [27]: Proposes a neighborhood-based neural collaborative filtering recommendation model, which is an algorithm that combines neighborhood-based collaborative filtering with a neural system to integrate the hiddeninformation in the neighborhood into the neural collaborative filtering, which contains the history of interactions between the item and the user, in order to make recommendations.

NGCF [28]: proposes a collaborative filtering algorithm based on graph convolution neural network, by defining the graph convolution in the null domain, the feature information generated by multiple iterations of the interaction information between the user and the item, and defined as the higher-order collaborative information, and the resulting higher-order features are cross-invariationally accumulated to make recommendations.

LightGCN [29]: the LightGCN recommendation algorithm is a simplification of the NGCF recommendation algorithm in terms of information transfer rules, removing the learnable parameters and nonlinear transformations in the traditional graph convolutional neural networks that have a negative impact on the results.

4.4. Experimental Setup

In the experiments, the dataset is randomly divided on the user dimension according to the ratio of 8:1:1 to construct the training, validation, and test sets; the parameters of all the models are as follows: the vector embedding dimension d is set to 64, the learning rate (lr) is set to 0.001, the batchsize size is set to 2048, the size of the L2 regularization coefficient λ is set to 1E- 4, weight decay is 0, and other hyperparameters are consistent with the original paper or code by default. In this paper, Adam is used as the optimizer, which can solve the gradient vanishing problem to some extent.

4.5. Experimental Results and Analysis

4.5.1. Comparison with benchmark algorithms

In this paper, two datasets, Movielens-1M and Movielens-100K, are used for experimental analysis, and the experimental results comparing this model with other baseline models are shown in Table 3, and the following conclusions are obtained:

Table 3. Comparative experimental results

mould	Movielens-1M			Movielens-100K		
	R@10(%)	N@10(%)	P@10(%)	R@10(%)	N@10(%)	P@10(%)
IRACF	13.30	12.48	8.67	12.15	9.41	6.34
MF	20.39	29.72	23.49	13.60	18.61	13.65
NNCF	21.13	31.58	26.20	18.42	22.03	16.91
NGCF	22.91	33.94	26.32	19.77	25.30	18.08
LightGCN	24.78	36.77	28.08	22.73	29.61	21.76
GCSCF	25.47	37.93	28.77	23.26	30.11	21.84

The GCSCF model in Movielens-1M dataset improves 12.17% and 5.08% in Recall metrics, 25.45% and 8.21% in NDCG metrics, and 20.1% and 5.28% in Precision metrics, respectively, than the traditional recommendation models IRACF and MF. The performance of the traditional recommendation model on both datasets is significantly lower than that of other models, indicating that the combination of the traditional collaborative filtering's recommendation algorithm and graph convolutional neural network effectively mitigates the data sparsity problem and the cold-start problem compared with the traditional recommendation model.

The GCSCF model in the Movielens-1M dataset improves over the NNCF, NGCF, and LightGCN models in the Recall metric by 4.34%, 2.56%, and 0.69%, in the NDCG metric by 6.35%, 3.99%, and 1.16%, and in the Precision metric by 2.57 %, 2.45 %, and 0.69 %, respectively. It illustrates that the GCSCF model effectively increases the performance of the recommendation algorithm by considering the similarity between items and aggregating similar items. Comparing to the NGCF model effectively enhances the accuracy of the model as well as alleviates the problem of long training time of collaborative filtering recommendation based on graph neural networks compared to traditional collaborative filtering algorithms due to redundancy of model parameters and excessive number of layers. Compared with the LightGCN model, the traditional similarity computation quickly mines the higher-order hidden information between similar items and uses the weight assignment based on the item scores to improve the recommendation efficiency to a certain extent.

Through the above analysis, there are two main factors for the good performance of the GCSCF model. First, the traditional similarity calculation can quickly mine the higher-order hidden information between similar items to enhance the model accuracy. Second, the feature vectors are propagated using a graph convolutional neural network to aggregate the localized information, and the weights are normalized and redistributed. This further enhances the ability of the model learning nodes to embed the propagated features and improves the learning effect of user preferences.

4.5.2. Ablation experiment

In order to verify the effectiveness of score-based preliminary aggregation in convolutional embedding propagation, ablation experiments are used to remove or change the GCSCF model by modules and design two ablation models to compare with this model, and the following conclusions are obtained:

GCSCF-a: predicts the final result by removing the similarity aggregation operation based on ratings, doing weight normalized embedding propagation only for item i and user u, and then final feature vectors doing inner product operation.

GGCSCF-s: By the operation of weight normalization embedding removal, item i is subjected to score-based aggregation, and then convolutional embedding is performed to get the feature vector high-order representation, and finally the feature vector is subjected to inner product operation to predict the final result.

Table 4. Ablation experiments

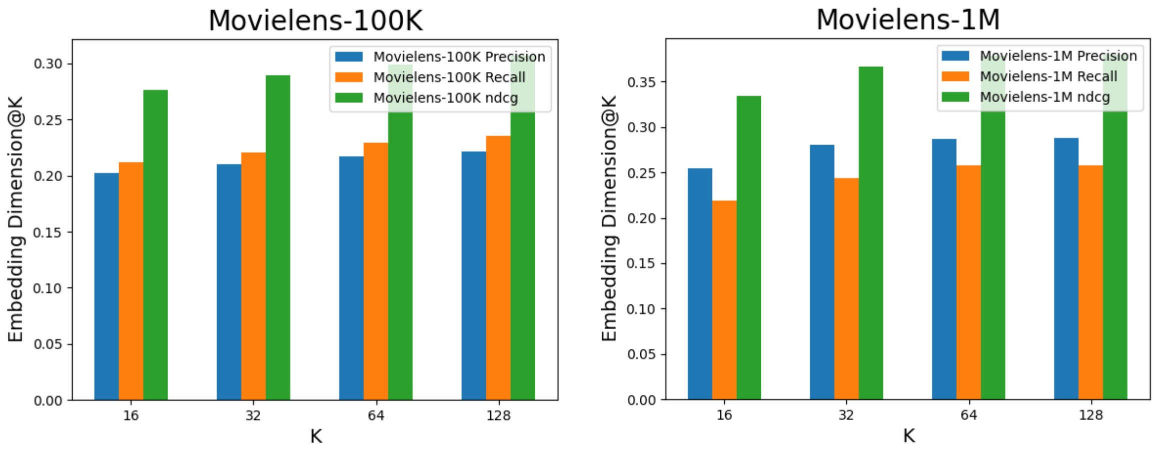
mould	Movielens-1M			Movielens-100K		
	R@10(%)	N@10(%)	P@10(%)	R@10(%)	N@10(%)	P@10(%)
GCSCF	25.47	37.93	28.77	23.26	30.11	21.84
GCSCF-a	25.24	37.43	28.53	22.36	29.26	21.54
GCSCF-s	25.36	37.78	27.92	22.95	29.64	21.81

According to Table 4, it can be seen that GCSCF is higher than other ablation models in all the indexes, which indicates that aggregating the dataset through the score similarity algorithm can mine the higher-order information hidden in the similar items. In addition the weighted normalized embedding can effectively eliminate the noise generated by dataset aggregation.

4.5.3. Hyperparametric analysis

(1) The effect of model embedding dimensions

The effect of different embedding dimensions on recommendation results is investigated by setting the range of embedding dimension K to {16, 32, 64, 128}, and the experiments are conducted on Movielens-100K and Movielens-1M datasets, respectively, with other parameters kept constant. The evaluation metrics are NDCG@K, Recall@K and Precision@K:



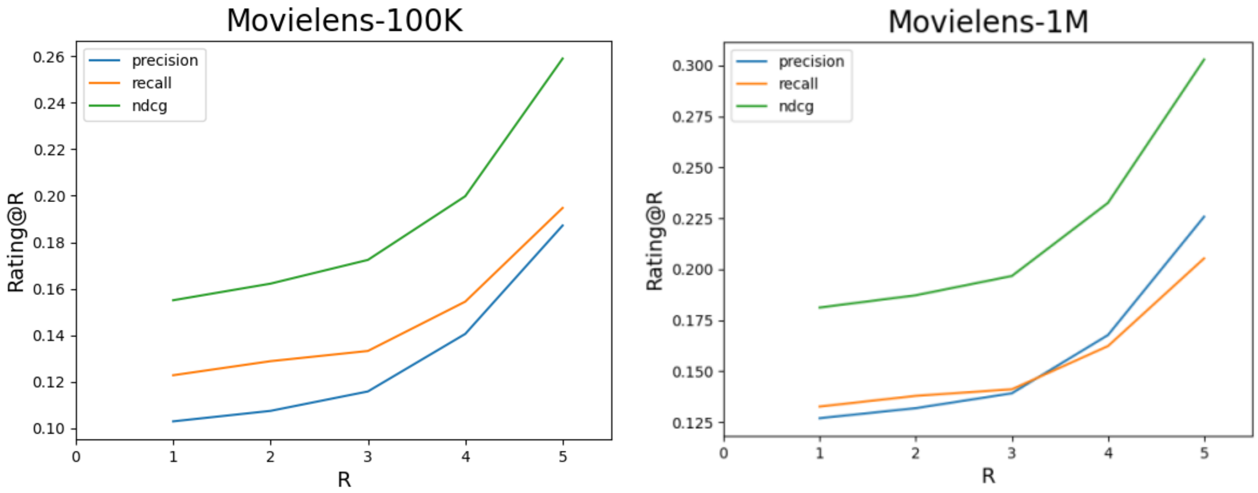
(a) Movielens-100K dataset (b) Movielens-1M dataset

Figure 2. Effect of embedding dimensions on modeling

As can be seen from Figure 2, the model performance index is best when the embedding dimension is 64. Therefore, it shows that when the embedding dimension is larger, the model will have overfitting phenomenon; when the embedding dimension is smaller, the model is not comprehensive enough to express the features and thus reduce the performance index.

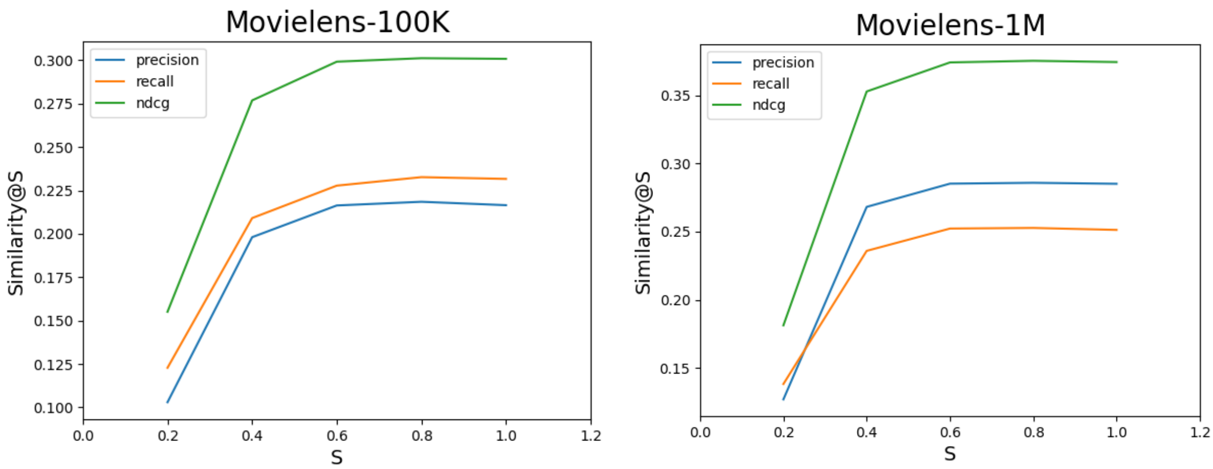
(2) Effects of model item scores and similarity

The effects of rating R and similarity S on recommendation results are investigated by setting the range of rating R to {1, 2, 3, 4, 5} and the range of similarity S to {0.2, 0.4, 0.6, 0.8, 1.0}, and aggregating items with greater than different ratings and different similarities, respectively, on Movielens-100K and Movielens-1M Experiments were conducted on the datasets with other parameters kept constant. The evaluation metrics are NDCG, Recall and Precision, respectively:



(a) Movielens-100K dataset (b) Movielens-1M dataset

Figure 3. Impact of item scoring and similarity on modeling



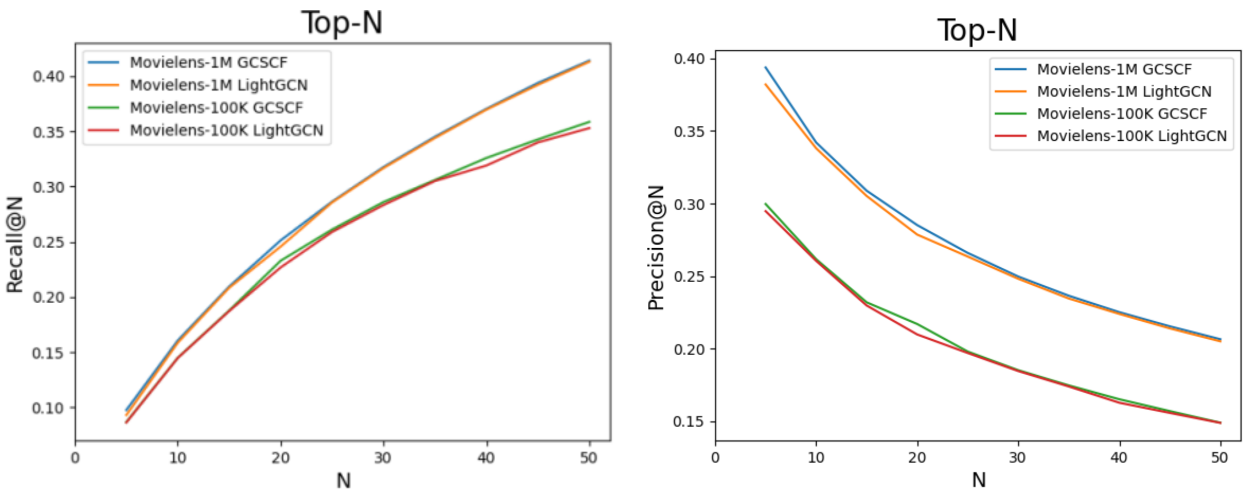
(a) Movielens-100K dataset (b) Movielens-1M dataset

Figure 4. Impact of project similarity on the model

From Fig. 3 and Fig. 4, it can be seen that when the rating R and similarity S are increased from 1 to 5 and from 0.2 to 1, respectively, the model GCSCF has the best performance when the rating is 5 and the similarity 0.8; when the rating and similarity are greater than 5 and 0.8, respectively, the performance index of the model GCSCF is a decreasing trend, so the model adopts the parameter of the rating of 5 and the similarity 0.8.

Model Top-k prediction experiment

Top-N recommendation experiments using the LightGCN model with the GCSCF model using two metrics, Recall@N and Precision@N, where the range of N is set to {5, 10, 15, 20, 25, 30, 35, 40, 45, 50} to investigate the effect of the number of recommendations on the model on the Experiments on Movielens-100K and Movielens-1M datasets.



(a) Recall@N (b) Precision@N

Figure 5. Top-k recommendation effect

By observing Fig. 5, we know that the GCSCF model outperforms the LightGCN model in overall performance. There is not much difference between the GCSCF model and the LightGCN model when N is less than 20, and the GCSCF model is significantly better than the LightGCN model when N is greater than 20, indicating that the GCSCF model performs better in the face of a wide range of recommendations.

5. CONCLUDING REMARKS

Aiming at the current collaborative recommender system based on graph convolutional neural network for the similar items between the existence of some high-order hidden information is difficult to capture as well as the existence of facing parameter redundancy, training time is too long and other problems, this paper proposes a collaborative filtering algorithm and graph convolutional neural network algorithm combined with a hybrid recommender algorithm (GCSCF). GCSCF through the similarity calculation based on the score of the similar items to the preliminary aggregation, then obtains the feature representations of users and items through weighted normalized embedding propagation with neighborhood aggregation layer, and finally performs matrix decomposition of the feature representations of users and items for recommendation, so as to improve the recommendation accuracy. However, it lacks the mining of the existed higher-order information between similar users and the utilization of other aspects of the items, such as users' comments and pop-ups. In future research, combining with user personalized interests and item multimodality will be considered.

REFERENCES

[1] STANLEY A J, CLIPSHAM P S.Information overloadmyth or reality? [C]//IEEE Colloquium on IT Strategies for Information Overload, 1998.

[2] LIU Jianguo, ZHOU Tao, WANG Binghong. Research progress of personalized recommendation system [J]. Progress in Natural Science, 2009, 19(01):1-15.

[3] W. T. Zhao, H. H. Tian, T. T. Feng, et al. Collaborative filtering algorithm incorporating similarity and prescreening patterns [J]. Computer Science and Exploration, 2023, 17(1): 217-227.

[4] KARABADJI N E I, BELDJOUDI S, SERIDI-BOUCHELAGHEM H, et al. Improving memory-based user collaborative filtering with evolutionary multi-objective optimization [J]. Expert Systems with Application, 2018, 98 (May): 153-165.

[5] LIU Y, WANG S L, ZHANG J F, et al. A neural collaborative filtering method for identifying miRNA- disease associations [J]. Neurocomputing, 2021, 422: 176-185.

[6] CHEN H, QIAN F, CHEN J, et al. Attribute-based neural collaborative filtering [J]. Expert Systems with Applications, 2021, 185(4):115539.

[7] ANWAR T, UMA V, SRIVASTAVA G. Rec-CFSVD ++: implementing recommendation system using collaborative filtering and singular value decomposition (SVD)++ [J]. International Journal of Information Technology and Decision Making, 2021(4): 1-19.

[8] HAN X, WANG Z, XU H J.Time-weighted collaborative filtering algorithm based on improved mini batch Kmeans clustering [J]. Advances in Science and Technology, 2021, 105: 309-317.

[9] Guo Yanhong, Deng Guishi. A hybrid recommendation algorithm for project cold-start in collaborative filtering systems [J]. Computer Engineering, 2008, 34(23):11-13.

[10] Guo Y, Wang M, Li X. An Interactive Personalized Recommendation System Using the Hybrid Algorithm Model [J]. Symmetry, 2017, 9(10):216-216.

[11] BRONSTEIN M M, BRUNA J, LECUN Y, et al. Geometric deep learning: going beyond euclidean data [J]. IEEE Signal Processing Magazine, 2017, 34(4):18-42.

[12] KIPF T N, WELLING M. Semi-supervised classification with graph convolutional networks [J]. arXiv: 1609.02907, 2016.

[13] Chami I, Ying Z, Ré C, et al. Hyperbolic graph convolutional neural networks [J]. Advances in neural information processing systems, 2019, 32: 4865-4876.

[14] Wang X, He X, Wang M, et al. Neural graph collaborative filtering [C]//Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. Paris, France: ACM, 2019: 165-174.

[15] Z. -D. Zhao and M. -s. Shang, "User-Based Collaborative-Filtering Recommendation Algorithms on Hadoop," 2010 Third International Conference on Knowledge Discovery and Data Mining, Phuket, Thailand, 2010, pp. 478-481, doi: 10.1109/WKDD.2010.54.

[16] Wu Jian-wei, Yu Jiu-hua and Yu Xiao-hong, "An item-based weighted collaborative recommended algorithm based on screening users' preferences," 2010 International Conference on Computer Application and System Modeling (ICCASM 2010), Taiyuan, China, 2010, pp. V8-161-V8-164, doi: 10.1109/ICCASM.2010.5619326.

[17] H. Liu, "Implementation and Effectiveness Evaluation of Four Common Algorithms of Recommendation Systems - User Collaboration Filter, Item-based Collaborative Filtering, Matrix Factorization and Neural Collaborative Filtering," 2022 International Conference on Cloud Computing, Big Data Applications and Software Engineering (CBASE), Suzhou, China, 2022, pp. 224-227, doi: 10.1109/CBASE57816.2022.00049.

[18] X. Pu and B. Zhang, "Clustering collaborative filtering recommendation algorithm of users based on time factor," 2020 Chinese Control and Decision Conference (CCDC), Hefei, China, 2020, pp. 364-368, doi: 10.1109/CCDC49329.2020.9164315.

[19] R. G. Lumauag, A. M. Sison and R. P. Medina, "An Enhanced Recommendation Algorithm Based on Modified User-Based Collaborative Filtering," 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS), Singapore, 2019, pp. 198-202, doi: 10.1109/CCOMS.2019.8821741.

[20] W. Zhao and H. Zhang, "The improved item-based clustering collaborative filtering algorithm based on Hadoop," 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 2017, pp. 2416-2419, doi: 10.1109/IAEAC.2017.8054456.

[21] S. Wei, N. Ye, S. Zhang, X. Huang and J. Zhu, "Item-Based Collaborative Filtering Recommendation Algorithm Combining Item Category with Interestingness Measure," 2012 International Conference on Computer Science and Service System, Nanjing, China, 2012, pp. 2038-2041, doi: 10.1109/CSSS.2012.507.

[22] N. Ifada, M. K. Sophan, M. N. Fitriantama and S. Wahyuni, "Collaborative Filtering Item Recommendation Methods based on Matrix Factorization and Clustering Approaches," 2020 10th Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS), Malang, Indonesia, 2020, pp. 226-230, doi: 10.1109/EECCIS49483.2020.9263450.

[23] Jingjing Zhang. Research and implementation of collaborative filtering recommendation algorithm based on matrix decomposition [D]. Nanjing University of Posts and Telecommunications, 2023. DOI:10.27251/d.cnki.gnjdc.2023.001011.

[24] Wu Z, Pan S, Chen F, et al. A comprehensive survey on graph neural networks [J]. IEEE Transactions on Neural Networks and Learning Systems, 2020, 32(1): 4-24.

[25] TANG J, QU M, WANG M, et al. LINE: large- scale information network embedding [C]//Proceedings of the International World Wide Web Conferences Steering Committee, 2015.

[26] Lee J B, Rossi R A, Kong X, et al. Higher-order graph convolutional networks [C]//Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA: PMLR, 2019: 646-655.

[27] BAI T, WEN J R, ZHANG J, et al. A neural collaborative filtering model with interaction-based neighborhood [C]//Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017.

[28] WANG X, HE X, WANG M, et al. Neural graph collaborative filtering [C]//Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019: 165-174.

[29] HE X, DENG K, WANG X, et al. LightGCN: simplifying and powering graph convolution network for recommendation [C]//Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020:639-648.

[30] Z. Li, M. Huang and Y. Zhang, "A Collaborative Filtering Algorithm of Calculating Similarity Based on Item Rating and Attributes," 2017 14th Web Information Systems and Applications Conference (WISA), Liuzhou, China, 2017, pp. 215-218, doi: 10.1109/WISA.2017.35.

[31] RENDLE S, FREUDENTHALER C, GANTNER Z, et al. BPR: Bayesian Personalized Ranking from Implicit Feedback [J]. CoRR. 2012, abs/1205.2618.