ITC 1/51
Information Technology and Control
Vol. 51/ No. 1/ 2022
pp. 18-31
DOI 10.5755/j01.itc.51.1.28720

A Graph Convolution Network Based on Improved Density Clustering for Recommendation System

Received 2021/03/25 | Accepted after revision 2021/11/16

http://dx.doi.org/10.5755/j01.itc.51.1.28720

HOW TO CITE: Li, Y. (2022). A Graph Convolution Network Based on Improved Density Clustering for Recommendation System. *Information Technology and Control*, 51(1), 18-31. https://doi.org/10.5755/j01.itc.51.1.28720

A Graph Convolution Network Based on Improved Density Clustering for Recommendation System

Yue Li

School of Information, Guangdong Communication Polytechnic, Guangzhou, China;
e-mails: liyue@gdcp.edu.cn, liyuely29@yeah.net

Corresponding author: liyue@gdcp.edu.cn; liyuely29@yeah.net

Recommendation systems have been widely used in various applications to solve information overload and improve user experience. Traditional recommendation algorithms mainly used Euclidean data for calculation and abandoned the graph structure features in user and item data. Aiming at the problems in the current recommendation algorithms, this paper proposes an improved user density clustering method and extracts user features through optimized graph neural network. Firstly, the improved density clustering method is used to form the clustering subgraph of users based on the influence value of users. Secondly, the user data and item data features of cluster subgraph are extracted by graph convolution network. Finally, the features of cluster subgraphs are processed by global graph convolution network and the recommendation results are generated according to the global graph features. This model not only improves the efficiency of decomposing large graph into small graph through the improved user density clustering algorithm, but also extracts the features of user groups through graph convolution neural network to improve the recommendation effect. The experiment also proves the validity of this model.

KEYWORDS: density clustering, graph collapse, information overload, graph convolution network, recommendation system.

1. Introduction

The rapid growth of Internet information not only brings users rich data, but also makes users face the problem of data overload. Recommender system is a system applied to many online services such as e-commerce websites, social media, news information and other systems. By calculating and filtering user data, item data, interaction behavior, the recommendation system forms a result that meets the user's personalized needs, thus enhancing the user's stickiness and improving the system's efficiency [4, 37]. Traditional recommendation algorithms can be classified into three categories: content-based recommendation, collaborative filtering recommendation and hybrid recommendation [33]. Content-based recommendation method needs to extract the user's preference features and the feature data of items, and the recommendation result greatly depends on the selection of feature information [21]. The collaborative filtering method uses the similarity between users to discover the similarity of preferences among different users. However, the quality of the model depends on the accuracy and integrity of user rating data, which is prone to data sparseness and cold start problems [20]. The hybrid recommendation method combines different recommendation models, including feature fusion, algorithm fusion, and result fusion [5]. With the development of artificial intelligence, deep learning has been widely used in various fields and has achieved good application effects. The recommendation model based on deep learning technology can effectively fuse multi-source heterogeneous data, and the result does not depend on manually selected features. It can realize the end-to-end training from multi-source heterogeneous data to prediction [36]. Through the explicit and implicit features of user data, the recommendation model based on deep learning learns the nonlinear multi-level abstract expression of data, thus improving the recommendation performance [34].

No matter the traditional linear model or neural network model, Euclidean structure data is the main data to be processed. However, in the real world, much data is generated from non-Euclidean structure data. In the recommendation system, there is a complex graph network structure, which is formed by the fusion of various network data, such as the social relationship network between users, the evaluation data network of users to projects, and the hierarchical network data between projects. Because there is no fixed relationship and position between graph data, and the structure of graph node is not uniform, the current neural network model is unsuitable for processing graph data [7].

In order to solve this problem, we propose a recommendation algorithm based on graph clustering around the graph convolutional neural network. Firstly, the users' data in the recommendation system are clustered based on the user's influence, and the large graph structure data is collapsed into a graph containing K cluster collapse subgraph nodes. Secondly, use the graph convolutional network to convolve the cluster collapsed subgraphs to obtain the local features. Finally, global graph convolution is performed on the obtained subgraph features to obtain global graph data features and generate recommendation results. The model proposed in this paper can make full use of the graph structure data in the recommendation system to extract features. The method of dividing super-large-scale data into k subgraphs by graph collapse clustering in the model also solves the problems of huge users and data processing in the recommendation system. Experiments also prove the effectiveness of our model.

The rest of the paper is organized as follows. Section 2 introduces the work related to graph convolutional network and user density clustering. Section 3 details the model structure and implementation proposed in this paper. We compare the performance of the proposed method with the baseline method and discuses some optimization problems in this model through experiments in section 4. Finally, section 5 concludes this paper with potential future directions.

2. Related Work

Graph convolutional neural network is a generalization of convolution neural network in a graph structure, which can learn node features and structure features end to end at the same time [30]. Compared with the traditional convolution network, the graph convolution network has the same properties. The convo-

lution operator in the graph convolutional network is applicable to each node, and the operator is shared everywhere on different nodes [26]. The receptive field of the model is proportional to the number of layers. At the beginning of the convolution, each node contains the information of the direct neighbors. When the second layer of convolution is calculated, the information of the second-order neighbors can be included, so that the information involved in the calculation is more sufficient [3]. The receptive field of the model is proportional to the number of convolutional layers. The more convolutional layers, the more information involved in the operation. Graph convolution network also has the characteristics of deep learning [16]. For example, graph convolution network has a hierarchical structure, and features are extracted layer by layer. Graph convolution network increases the expression ability of the model through nonlinear transformation, and it can realize end-to-end training without defining any rules.

Although graph convolutional networks have very powerful capabilities, there are some limitations in constructing recommendation systems based on graph structures. In the recommendation system, the number of users is often huge. For example, the number of Amazon users exceeds 100 million [9], and the registered users' number of Douban exceeded 10 million [32]. There are some problems in constructing graph network based on tens of millions users, such as huge data processing and high calculation consumption. Because of the irregularity of the graph structure, all data needs to be recalculated when new user data is added. In order to solve the problems in the calculation of graph structure data, we propose a graph-based collapse convolution model. By collapsing the huge graph structure in the recommendation system into subgraphs based on user density, the large graph is divided into k subgraphs. We extract the data features of cluster collapse sub graph by graph convolution network, and then obtain the global features by graph convolution of subgraph super nodes from the perspective of global graph structure, and predict the user's score to get the recommended results. The detailed method will be introduced in section 3.

In our model, users need to be collapsed clustering based on user density. As the most popular clustering algorithm, K-means algorithm was first adopted by Macqueen [19] in 1967. Compared with other clustering algorithms, K-means algorithm is widely used in clustering because of its good results and simple idea. K-means algorithm generally uses the distance between data to measure the similarity. The similarity is inversely proportional to the distance between data. The greater the similarity, the smaller the distance. The algorithm needs to specify the original number of clusters and original cluster centers in advance, and then update the location of cluster centers according to the similarity between data objects and cluster centers, so that the Sum of Squared Error (SSE) of clustering decreases continuously [15]. When SSE no longer changes or the objective function converges, clustering is finished and the result is obtained. K-means algorithm is simple and easy to use, but it also has serious shortcomings because it needs to determine the number of cluster's k in advance and select the initial cluster center randomly [14]. In view of these deficiencies, scholars have put forward different improved algorithms. Rezaee et al. [24] proposed that the optimal value of k is in the range of $(1, \sqrt{n})$, and n is the size of data set. Han [8] proposed an improved algorithm to find the initial clustering center based on density, and introduced the concept of density parameter, which made the clustering test result on iris data set increase to 83.33%.

Different from the data in other systems, one of the important data in the recommendation system is user's information. In Chinese culture, there is a proverb for users, "Birds of a feather flock together", which means that in the recommendation system, users have the attribute of group gathering. This intrinsic connection between users will make users form density clustering [17]. Therefore, we can select the clustering center by clustering density, and improving the convergence efficiency of the algorithm.

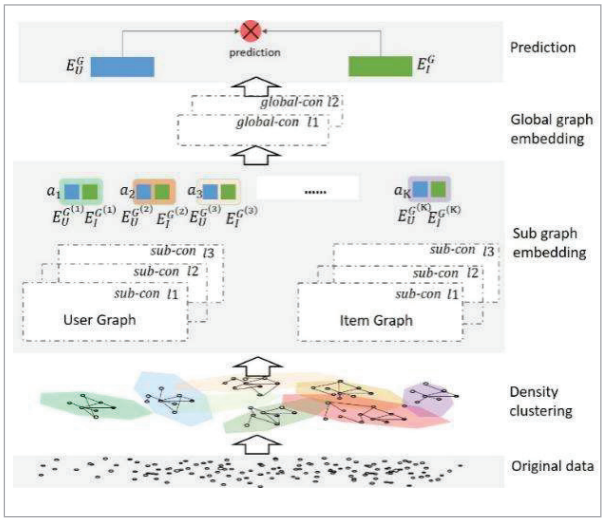
In user clustering, there is a data which can reflect the density of surrounding users and represent the characteristics of the surrounding nodes. We call it the cluster center node. Compared with other nodes, cluster center nodes have two properties. One property is that the density of cluster center nodes is very high, surrounded by neighboring nodes whose density is not higher than it. The density of common nodes in the cluster is not higher than the center nodes [22]. The other property is that the distance between the cluster central node and other cluster central nodes with higher density is larger. Rodriguez and Laio

[25] proposed the definition of a cluster center node in 2014, a new clustering algorithm named Density Peaks Clustering (DPC) algorithm. The biggest advantage of this algorithm is the fact that it doesn't need to determine the number of clusters in advance, and it can find and process arbitrary data sets. Therefore, users in the recommendation system can be clustered based on density, and k subgraphs of users can be formed. Users have various attributes, such as the original attributes, including gender, age, native place, educational background, etc., and implicit attributes include short-term and long-term interests. In this paper, the model comprehensively calculates the data of user influence and forms density clustering based on user influence.

3. Proposed Model

In this section, we proposed a graph convolutional network recommendation model based on graph density collapse (GDC-GCN). Figure 1 portrays the overall architecture of our model.

Figure 1
Architecture of proposed model



The model consists of four layers. The first layer is the user density clustering collapse layer. Through density clustering based on user influence, the original input data are collapsed into a collapsed graph structure containing k subgraphs. The second layer is subgraph convolution network layer, which gets the features of sub-graph through graph convolution network. The third layer is the global graph convolution layer, through the graph convolution of the sub-graph super node. The features of the global graph are obtained. Finally, the prediction layer is used to calculate the items that may interest to the user through the obtained global graph structure features, and generate a recommendation list.

3.1. User Influence Rank

In the recommendation system, user data are an important information component for calculating recommendation results. According to the impact on other users, users can be divided into authoritative users and ordinary users. Authoritative users have a high influence on the surrounding uses, attracting additional users to gather around themselves. Thus, forming a dense user group around authoritative users. At the same time, compared with ordinary users, authoritative users will impact users in a further range. Thus, forming a larger user influence range than ordinary users. Based on this premise, users with greater influence can be found first to form a candidate set of cluster center nodes. At the same time, two different authoritative users represent two kinds of users with different characteristics. Therefore, the two users who become the central nodes of the cluster should have greater distance and higher dissimilarity.

Firstly, N users can be selected according to the influence of users to form the candidate data of central nodes. The influence of users is an implicit feature, which comes from the following aspects:

a The influence of user's basic attributes. In the recommendation system, the basic attributes of users mainly include some explicit feature values, such as account number, gender, age, education level, and explicitly filled in areas of interest, etc. In addition, some systems also give users some explicit grades. For example, users in Sina Weibo can be ordinary users without any authentication, ordinary users with identity authentication, officially authenticated big V users, or users with self-media authentication. The integrity of the basic attributes of users and the trust given by the platform constitutes the influence of the basic attributes of users. The user's influence based on basic attributes is recorded as Au.

b The influence of users' positive behavior. The user's active behavior in the system will also increase the user's influence on the surrounding users and expand their influence value. For example, active users who log in frequently and post topics will have a greater impact on other users than silent users. User's active behavior refers to the behavior initiated by the user without social interaction with others as the main purpose, such as user logging in, posting posts, evaluating items, etc. Therefore, the user 'active behaviors such as logging in, publishing, visiting, commenting, forwarding, and replying to topics constitute the behavior influence of users, which is recorded as *Bu*.

c The influence of users' social behavior. There will be interaction between users in the recommendation system, and the influence of this interaction is higher than the influence of the user's own attribute value and the user's active behavior. Users' social behaviors include visiting, praising, paying attention, commenting, sending messages, forwarding or blocking others. The societal behavior of users is directional. For instance, a user pays attention to user B, but user B does not necessarily pay attention to user A. Similarly, a user A gives B a 'like', and B does not necessarily like A's posts. When considering the user's influence, the influence reflected by the social behavior sent by users is smaller than that reflected by the social behavior received by users. For example, posts posted by high-impact users with many fans in blog users will always get more praise, but such users may not praise the content of other users frequently. Therefore, when calculating the user's social behavior influence, what needs to be calculated is the social behavior influence data brought by the user as the receiver, that is, the in-degree of the node in the user's social relationship graph, which is recorded as *Su*.

Based on the influence data of these aspects, the formula for calculating the total influence of any user u in the system is as follows:

$$E_u = \alpha A_u + \beta B_u + (1-\alpha-\beta)S_u . \quad (1)$$

In the formula, *Au* represents the attribute influence of the user U, *Bu* represents the active behavior influence and *Su* represents the social behavior influence. For a system with N users, the users with the top $2\sqrt{n}$ influence are selected to form a cluster center candidate set.

3.2. Cluster Collapse Based on User Influence Density

For the user data in cluster center candidate set, the surrounding density of the user i denoted as ρi, the distance between the user u_i and u_j denoted as $dist(u_i, u_j)$ which can be calculated in Euclidean distance and formulated as follows:

$$dist(u_i,u_j) = \sqrt{\sum_{k=1}^{n}(u_{ik}-u_{jk})^2}, \quad (2)$$

where k is the dimension of user's feature. If the distance between the users is higher, the similarity between the users is lower, so the distance between the user and himself is 0. The larger the distance is, the lower the contribution of adjacent users to the density calculation of user I. To prevent outliers from interfering with the density data when obtaining the distance, the calculated user distance should be normalized. In this paper, the Z-score [6] normalization method is used to normalize the distance data. It can be formulated as follows:

$$dist(u_i,u_j)^* = \frac{(dist(u_i,u_j)-\mu)}{\sigma}, \quad (3)$$

where μ denotes the average value of the users and σ denotes the standard deviation of the users of the user sample data. The density calculation formula for user i is as:

$$\rho_i = \sum_{u_i \in U} D(dist(u_i,u_j)^* - d_c) . \quad (4)$$

Combining Formula 2 and Formula 3, Formula 4 can also be expressed as:

$$\rho_i = \sum_{u_i \in U} D(\frac{\sqrt{\sum_{K=1}^{n}(u_{ik}-u_{jk})^2}-\mu}{\sigma} - d_c) . \quad (5)$$

1The density contribution of user i to himself is 1, and the density contribution of other users to user i is calculated by Gaussian function as follows:

$$D(i,j) = \begin{cases} 1, \ if \ i=j \\ exp(-(\frac{\sqrt{\sum_{K=1}^{n}(u_{ik}-u_{jk})^2}-\mu}{\sigma})/d_c)^2), \ if \ i \neq j \end{cases} . \quad (6)$$

The cut-off value of node distance is d_c, which can also be regarded as a super parameter. The value of this parameter will not only affect the selection of the cluster center, but also affect the edge division of clustering results. Thus, the choice of cut-off distance is extremely important. Rodriguez and Laio [25] proposed a method so that the average number of neighbors of each sample point accounts for approximately 1% to 2% of the total number of sample points in the dataset. Wu et al. [28] proposed an algorithm to dynamically adjust the truncation distance by Gini coefficient. Bai et al. [2] proposed a truncation algorithm based on the nearest shared neighbor. Liu et al. [18] used the idea of k nearest neighbors to calculate the truncated distance. Wang et al. [27] proposed an optimal truncation distance extraction method based on the data field theory and information entropy theory. The optimal truncation distance value was obtained by calculating the potential energy entropy of the original data set in the data field. Xie et al. [31] proposed an algorithm based on the peak density of K-nearest neighbors to eliminate the influence of the selection of truncation distance on the clustering results. Rodriguez and Laio [25] pointed out that when the data set is large, the clustering results of DPC algorithms are less affected by the truncation distance, and vice versa. In this paper, the data quantity decreases exponentially after calculating the user influence to form the candidate cluster center set. Combined with other reference methods, the truncation distance in this paper adopts the average distance between candidate clusters in the candidate sample set of the cluster center. Combined with the methods of other references, the truncation distance is calculated by using the average distance between the candidate cluster centers in the set and formulated as follows:

$$d_c = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} dist(u_i,u_j)^*. \tag{7}$$

The user density is calculated and sorted in sequence for the user data in the cluster center candidate set. According to the sorting size, the first half of the data is chosen as the initial center value for the next K-means cluster. Compared with the size of the original data set, the size of the initial center set is $\sqrt{n}$.The data in this set are used as the initial clustering center for K-means clustering, so as to form a more complete clustering subgraph of the original dataset. The algorithm is as follows.

Algorithm 1: Density Clustering Based on User Influence

Inputs: A candidate cluster center set containing k elements and an initial data set containing n users.

Output: allocate n users to k clusters.

1: Taking k elements in the candidate cluster center set as the initial cluster center, and calculate the distance from all data x to the cluster center C_i, with the formula as:

$$d(x,C_i) = \sqrt{\sum_{j=1}^{k}(x_j - C_{ij})^2}. \tag{8}$$

2: Assign data objects to the closest cluster with the highest similarity.

3: Calculate the average value of data in each cluster and update the cluster center, use the formula:

$$\overline{x_i} = \sum_{x\in C_i}\frac{x}{|C_i|}. \tag{9}$$

4: Repeat steps 2 and 3 until the SSE of the whole data converge, and the algorithm ends.

$$SSE = \sum_{i=1}^{k}\sum_{x\in C_i}|d(x,C_i)|^2 \tag{10}$$

After the algorithm is completed, k centroid density clusters are formed for users in the recommendation system, and k subgraphs are obtained.

The data in the recommendation system are represented and stored by graph structure, which is marked as graph G. After k-means density collapse, k clusters of users are formed. According to these k clusters, k subgraphs can be constructed and denoted as $\{\mathcal{G}^{(k)}\}_{k=1}^{K}$.

For the k subgraph $\mathcal{G}^{(k)}$, the number of nodes contained in the subgraph is denoted as N_k, and the cluster collapse matrix of the subgraph is denoted as S_k.

3.3. Graph Convolutional Network on Subgraph

The cluster collapse matrix S_k is convoluted by graph convolution network to form collapse expression of

subgraph. For the recommendation system, the most important task is to get the user's evaluation data of the project to determine the recommended items for users. At the same time, the purpose of density-based collapse of users in the model is to obtain different user groups. Therefore, when mapping the folding matrix of a subgraph, there is unnecessary to stack too many convolution layers. Graph convolution neural network has the characteristic of low-pass filtering [12 He]. Although low-pass filtering can make the signal smoother, too many convolution layers will make the signal converge more and more, resulting in over-smoothing. Therefore, the appropriate convolution layer should be selected in the model.

In the recommendation system, the data to be processed include user data and item data. For the cluster collapse subgraph $\mathcal{G}^{(k)}$, calculated the user embedding and the item embedding. The layer-l+1 embedding of use and item is formulated as follows:

$$e_u^{(l+1)} = Leaky\,\mathrm{Re}\,LU(W_1^{l+1}e_u^{(l)} + \\ + \sum_{i\in N_u}\frac{1}{\sqrt{|N_u||N_i|}}(W_1^{l+1}e_i^{(l)} + W_2^{l+1}(e_i^{(l)} \odot e_u^{(l)}))) \quad (11)$$

$$e_i^{(l+1)} = Leaky\,\mathrm{Re}\,LU(W_1^{l+1}e_i^{(l)} + \\ + \sum_{u\in N_i}\frac{1}{\sqrt{|N_u||N_i|}}(W_1^{l+1}e_u^{(l)} + W_2^{l+1}(e_u^{(l)} \odot e_i^{(l)}))). \quad (12)$$

The layer-0 embedding is the original input data of the graph. The user-item evaluation matrix is denoted as R and $R \in \mathbb{R}^{M\times N}$, where m is the number of users and n is the number of items. The adjacency matrix A can be constructed for the evaluation matrix of user items as:

$$\mathrm{A} = \begin{pmatrix} 0 & R \\ R^T & 0 \end{pmatrix}.$$

In the calculation of graphic volume, the convolution operator adopts the normalized Laplace matrix $\tilde{L}$ with the formula:

$$\tilde{L} = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}. \quad (13)$$

In the neural network of convolution graph, the convolution calculation formula of the layer-l is:

$$E^{(l)} = leaky\,\mathrm{Re}\,LU(\tilde{L}E^{(l-1)}W). \quad (14)$$

$E^{(0)}$ is the initial input data of the graph. According to the research of He et al. [11], multi-layer convolution is unnecessary and even will reduce the accuracy of the model. Therefore, the k-th cluster collapse subgraph is formed by only one layer of convolution. The user embedding $E_u^{G^{(k)}}$, while $u \in G^k$ and item embedding $E_i^{G^{(k)}}$ while $i \in G^k$ of the sub-graph $\mathcal{G}^{(k)}$ can be calculated as:

$$E_u^{G^{(k)}} = leaky\,\mathrm{Re}\,LU(W_1e_u^{(0)} + \\ + \sum_{i\in N_u}\frac{1}{\sqrt{|N_u||N_i|}}(W_1e_i^{(0)} + W_2(e_i^{(0)} \odot e_u^{(0)}))) \quad (15)$$

$$E_i^{G^{(k)}} = leaky\,\mathrm{Re}\,LU(W_1e_i^{(0)} + \\ + \sum_{i\in N_i}\frac{1}{\sqrt{|N_u||N_i|}}(W_1e_u^{(0)} + W_2(e_u^{(0)} \odot e_i^{(0)}))). \quad (16)$$

Because the user groups with different characteristics have different contributions to the scoring prediction task, in the global graph convolution network, it is necessary to set sub-graph weight for the features obtained by collapsing sub-graphs of different clusters. By synthesizing the features of subgraphs, we can get the user features E_U^G and item features E_I^G of the global graph structure.

$$E_U^G = a_1E_U^{G(1)} + a_2E_U^{G(2)} + \ldots + a_kE_U^{G(k)} \quad (17)$$

$$E_I^G = a_1E_I^{G(1)} + a_2E_I^{G(2)} + \ldots + a_kE_I^{G(k)}. \quad (18)$$

For a graph neural network with L layers, after the convolution of L layers is extracted, the user embedding and item embedding of each layer can be obtained. The embedding obtained at the last layer of the network is used to complete the prediction task, and the uses' score prediction for item i is denoted as $\hat{p}$ and can be calculated as:

$$\hat{p}_{u,i} = e_u^T e_i. \quad (19)$$

To learn model parameters, we optimize the pairwise Bayesian personalized ranking (BPR) loss [23], which has been intensively used in recommender systems. It considers the relative order between observed and unobserved user-item interactions. Specifically, BPR assumes that the observed interactions, which

are more reflective of a user's preferences, should be assigned higher prediction values than unobserved ones. The objective function is as follows.

$$Loss = \sum_{u=1}^{M} \sum_{(i,j)\in F} -\ln sigmod(\hat{p}_{u,i} - \hat{p}_{u,j}) + \lambda \left\| \Theta \right\|_2^2 . \quad (20)$$

While $F\{(i,j)|i \in (u,i) \neq 0, j \in (u,j)\}=0$, λ indicate $L2$ regularization, Θ denotes all trainable model parameters.

4. Experiments

4.1. Datasets

In order to evaluate the actual performance of our model, we select two public datasets for experiment. MovieLens [10] is collected by the GroupLens research team, which contains multiple users' rating data for multiple films. The dataset contains 69,878 users and 10,681 items, and 10,000,054 evaluation associations between users and items, each user evaluates at least 20 films. In the MovieLens dataset, there is no explicit influence data among users, so the user collapse clustering in the MovieLens in this paper uses an implicit relationship calculation.

According to Asghar [1], Yelp includes data such as users, businesses and users' comments on businesses. The experimental data selected in this paper include 31,668 user data and 1,561,406 user comments on 38,048 businesses. Each user give ratings in the range [0*;* 5]. Similarly, there is no explicit user influence data in Yelp. Therefore, it is necessary to obtain the user influence value through implicit conversion. Table 1 shows the statistics of these datasets.

Table 1
Statistics of the datasets

Datasets	User	Item	Rating	Density
MovieLens	69878	10681	100,000,054	0.01340
Yelp	31668	38048	1561406	0.00130

The data set is divided into training set and test set at the ratio of 8:2. The parameters in the model are randomly optimized by Adam algorithm. The learning rate is set to 0.001 and the dropout is set to 0.3. The maximum epoch is 200 with early stop design. The batch size is set to 512.

4.2. Baseline

We compare the proposed GDC-GCN model with the following baselines for recommender systems.

- UBCF [35] is a traditional collaborative filtering model. The model only considers users without considering item information. Thus, it is a user-based collaborative filtering recommendation algorithm.
- BPCF [13] is a collaborative filtering recommendation algorithms based on bipartite graph partitioning co-clustering. Firstly, users and items are constructed into a bipartite graph for co-clustering, which is then mapped to the low-dimensional feature space. Then, the proposed algorithm computes two types of improved similarities according to the clustering results and combines them. Based on the combined similarity, the user-based approach are adopted, respectively to predict for an unknown target rating and these prediction results are fused.
- NGCF [12] is a model that applies graph neural network to recommendation system. This model exploits the user-item graph structure by propagating embedding on it. This leads to the expressive modeling of high-order connectivity in user-item graph, effectively injecting the collaborative signal into the embedding process in an explicit manner.
- SocialGCN [29] is a model that combines the strength of graph convolution neural (GCNs) for modeling the diffusion process in social networks and the classical latent factor based models for capturing user-item preferences.

4.3. Experimental and Analysis

4.3.1. The Number of Layers in Graph Convolution Network

For clustering collapse subgraphs formed in the model, multi-layer GCN can be stacked to extract the features of subgraphs. However, because of its low-pass filtering characteristics, the graph convolution neural network will produce over-smooth after multi-layer convolution, which reduces the performance of the

model. Therefore, it needs to set an optimal number of convolution layers. The experimental data shows that in MovieLens, the highest evaluation index is obtained when the sub-graph convolution layers are 3, while in the Yelp dataset, the best sub-graph convolution layers are 2. However, no matter on MovieLens or Yelp datasets, it can be seen that with the increase of convolution layers, the evaluation indexes show a significant downward trend, indicating that with the deepening of the level, the model cannot obtain more abundant user and item features, but may reduce the quality of feature extraction due to excessive smoothing. On both MovieLens and Yelp datasets, with the increase of convolution layers, the evaluation indexes show a significant downward trend (Figure 2).

This shows that with the deepening of the hierarchy, the model can not only obtain more abundant features of users and objects, but may reduce the quality of feature extraction because of its over-smoothness. For the convolution of the global graph structure, it is not necessary to pile up too many graph convolutional layers, or even reduce the model performance.

For the convolution of the global graph, it does not need to pile up too many graph convolutional layers, or even reduce the model performance. It can be seen through experiments that the global graph convolution is more appropriate to set to 2 layers, as shown in Figure 3. In order to obtain better recommendation performance, the algorithm model in this paper uses

Figure 2
Experimental results of the influence of convolution layers of subgraphs

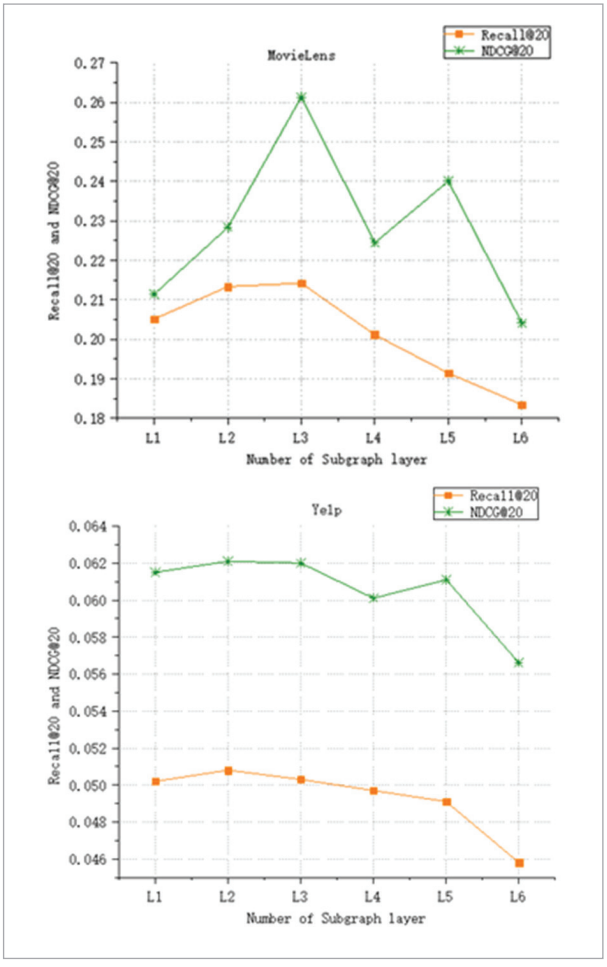
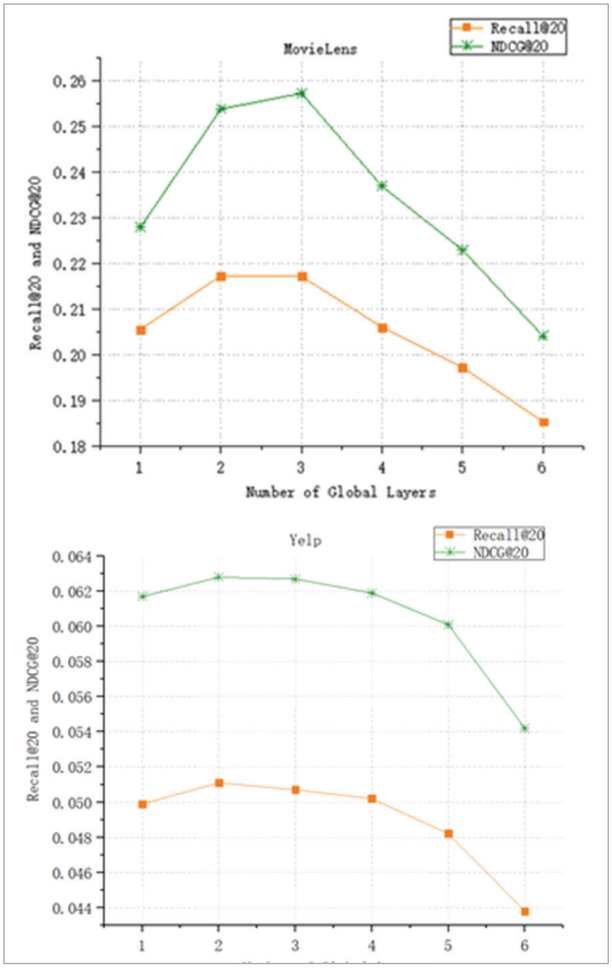


Figure 3
Experimental results of the influence of convolution layers of global graphs



a 3-layer sub-graph convolutional layer and a 2-layer global graph convolutional layer.

4.3.2. Cluster Collapse Subgraph Weights

In our model, when cluster collapse subgraphs are aggregated into a global graph structure, we set a weight parameter for each subgraph. Compared with the full-graph convolution model (GDC-GCN) without setting the weight of subgraphs (GDC-GCN-Sim), the results of the two models on different data sets are shown in Figure 4. As the number of graph convolutional layers increases, the model with sub-graph parameters can obtain higher NDCG and Recall than the model without sub-graph parameters, which indicate that setting weight parameters of subgraph have a certain ability of adjusting over-smooth.

4.3.3. Dropout

We can set the nodes dropout and edges dropout to prevent over-fitting. We can also set a dropout ratio on the super-node formed by the cluster collapse subgraph. Figure 5 (a) (b) plots the effect of node dropout and edge dropout on subgraph against different evaluation protocols on different datasets, and (c) (d) plots the effect of node dropout on global graph. Experiments show that dropout on nodes are effective on different data sets, but dropouts on edges may reduce model performance. The reason is that dropout of edges may lose effective neighbor node aggregation information, which has a negative impact on node feature extraction. At the same time, the dropout of nodes on the subgraph super-node cannot improve the performance of the model. The number of users

Figure 4

Experimental results of subgraph weights

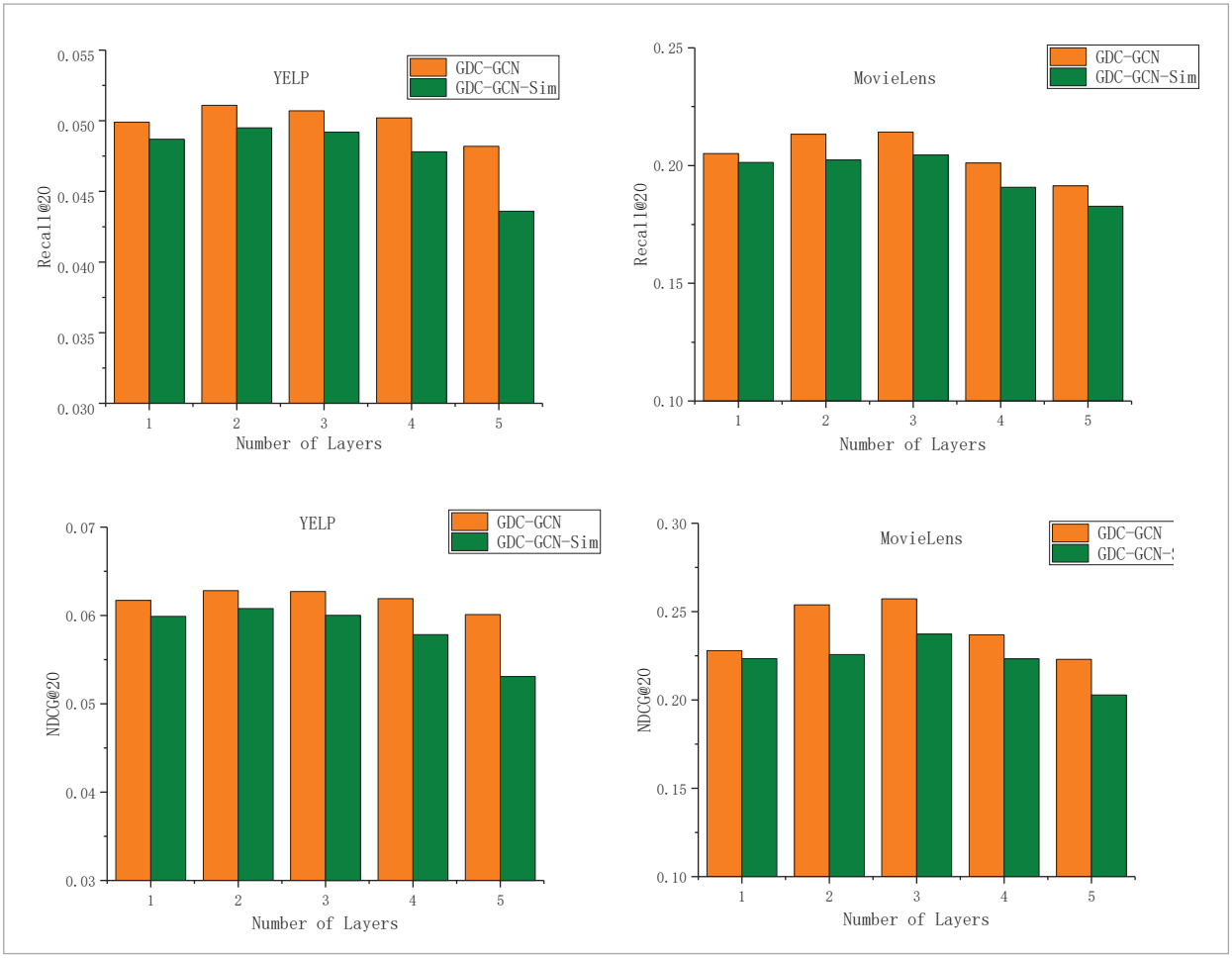
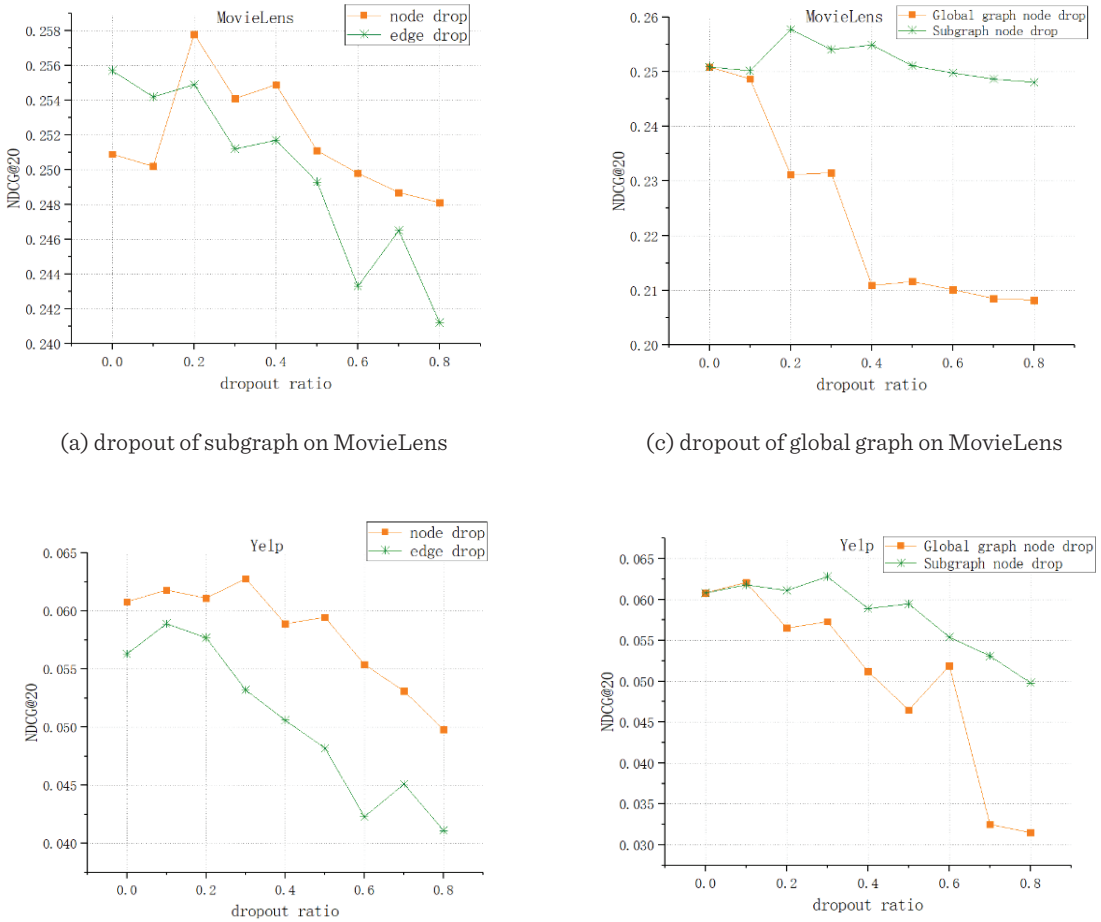


Figure 5
Experimental results of dropout ratio on different methods



(a) dropout of subgraph on MovieLens

(c) dropout of global graph on MovieLens

(b) dropout of subgraph on Yelp

(d) dropout of global graph on Yelp

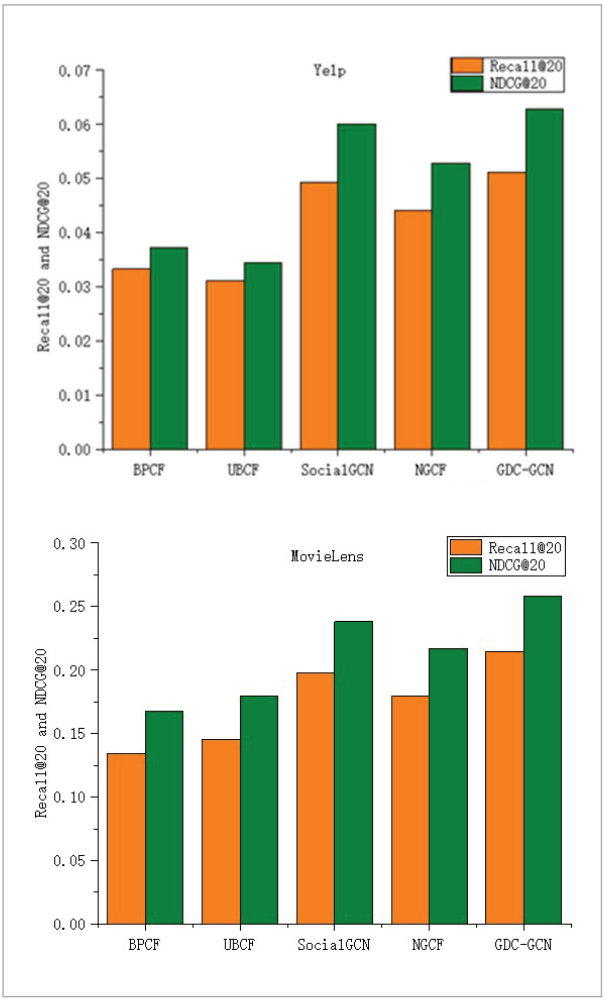
in the experimental data set is 69,878 and 31,668. The maximum number of subgraphs after collapse is 265 and 178. The dropout optimization effect of global graph nodes on small data is not obvious. However, whether node dropout on global graph on super large user data is useful to improve the performance of the model is worth further study.

4.3.4. Performance Comparison with Baseline

The model proposed in this paper adopts 3 layer cluster collapse subgraph convolution layer and 2 layer global graph convolution layer. The feature extraction of cluster collapse subgraph adopts node-based dropout, and the dropout ratio is 0.3. The different models are unified to generate a list of Top 20 recommendations.

Through comparison, it is found that the model proposed in this paper has a better performance in both Recall rate and NDCG indicators, indicating that the model proposed in this paper is applicable. Figure 6 plots the results.

Figure 6
Comparison with baseline



5. Conclusions

In this paper, we proposed a graph convolutional network recommendation model based on graph density collapse (GDC-GCN). It is a recommendation algorithm based on graph structure data, and proposes a collapse clustering method based on user influence density for the clustering relationship between users in the recommendation system. Through training of graph convolution network of clustering folding subgraph and global graph, more abundant data features are obtained.

In order to get the best recommendations, experiments have shown that with the increase of the number of layers, neither the sub-graph convolution layer nor the global convolution layer will improve the recommendation performance. Through experiments, it is found that the optimal number of sub-graph convolution network stacked layers is 3, and the optimal number of global convolution network stacked layers is 2. This paper also discussed the impact of node dropout and edge dropout on model performance, and pointed out that adding dropout to global graph convolution is not beneficial to model performance. At the same time, the experiment also showed that when the sub-graph super node constructs the global graph structure, by setting the sub-graph weight parameters, it can better utilize the preferences of the group users to produce recommendation results. Experiments on two public data sets also prove the effectiveness of this model.

For further work, we will continue to study the cluster optimization problem in cluster collapse subgraph and the performance impact of adding dropout in global graph structure on different size and sparsity datasets.

References

1. Asghar, N. Yelp Dataset Challenge: Review Rating Pre-diction. arXiv preprint arXiv:1605.05362, 2016. Accessed on September 21, 2020.
2. Bai, E., Luo, K., Luo, X. Peak Density Clustering Algo-rithm Combining Natural and Shared Nearest Neighbor. Journal of Frontiers of Computer Science and Technology, 2021, 15(5), 931-940. https://doi.org/10.3778/j.issn.1673-9418.2006060
3. Berg, R. V. D., Kipf, T. N., Welling, M. Graph Convolu-tional Matrix Completion. arXiv preprint arXiv:1706.02263, 2017. Accessed on July 23, 2019.
4. Bobadilla, J., Ortega, F., Hernando, A., Gutiérrez, A. Recommender Systems Survey. Knowledge-Based Systems, 2013, 46, 109-132. https://doi.org/10.1016/j.knosys.2013.03.012
5. Burke, R. Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interac-tion, 2002, 12(4), 331-370. https://doi.org/10.1023/A:1021240730564
6. Cheadle, C., Vawter, M. P., Freed, W. J., Becker, K. G. Analysis of Microarray Data Using Z Score Transfor-mation.

The Journal of Molecular Diagnostics, 2003, 5(2), 73-81. https://doi.org/10.1016/S1525-1578(10)60455-2

7. Davoudi, A., Chatterjee, M. Social Trust Model for Rating Prediction in Recommender Systems: Effects of Similarity, Centrality, and Social Ties. Online Social Networks and Media, 2018, 7, 1-11. https://doi.org/10.1016/j.osnem.2018.05.001
8. Han, L. A New Method of Determining the Optimal Class Number for K-Means. Modern Computer, 2013, 30, 12-15.
9. Haque, T. U., Saber, N. N., Shah, F. M. Sentiment Analysis on Large Scale Amazon Product Reviews. Proceedings of IEEE International Conference on Innovative Research and Development, (ICIRD 2018), Bangkok, Thailand, May 11-12, 2018, 1-6. https://doi.org/10.1109/ICIRD.2018.8376299
10. Harper, F. M., Konstan, J. A. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems, 2016, 5(4), 1-19. https://doi.org/10.1145/2827872
11. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M. LightGCN: Simplifying and Powering Graph Convolu-tion Network for Recommendation. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, (SIGIR 2020), Virtual Event, China, July 25-30, 2020, 639-648. https://doi.org/10.1145/3397271.3401063
12. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T. S. Neural Collaborative Filtering. Proceedings of the 26th International Conference on World Wide Web, (WWW 2017), Perth, Australia, April 3-7, 2017, 173-182. https://doi.org/10.1145/3038912.3052569
13. Huang, L., Ma, H., Li, N., Yu, L. A Collaborative Filtering Recommendation Algorithm Based on Bipartite Graph Partitioning Co-Clustering. Computer Engineering and Science, 2019, 41(11), 2040-2047. https://doi.org/10.3969/j.issn.1007-130X.2019.11.019
14. Huang, Z. Extensions to the K-Means Algorithm for Clustering Large Data Sets with Categorical Values. Data Mining and Knowledge Discovery, 1998, 2(3), 283-304. https://doi.org/10.1023/A:1009769707641
15. Jain, A. K. Data Clustering: 50 Years beyond K-Means. Pattern Recognition Letters, 2010, 31(8), 651-666. https://doi.org/10.1016/j.patrec.2009.09.011
16. Kipf, T. N., Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. arXiv preprint arXiv:1609.02907, 2016. Accessed on February 11, 2020.
17. Likas, A., Vlassis, N., Verbeek, J. J. The Global K-Means Clustering Algorithm. Pattern Recognition, 2003, 36(2), 451-461. https://doi.org/10.1016/S0031-3203(02)00060-2
18. Liu, Y., Ma, Z., Yu, F. Adaptive Density Peak Clustering Based on K-nearest Neighbors with Aggregating Strategy. Knowledge Based Systems, 2017, 133, 208-220. https://doi.org/10.1016/j.knosys.2017.07.010
19. MacQueen, J. Some Methods for Classification and Anal-ysis of Multivariate Observations. Proceedings of the Fifth Berkeley Symposium on Mathematical, Statistics, and Probability, Berkeley, Calif, USA, 1967, 1, 281-297.
20. Nakamura, A., Abe, N. Collaborative Filtering Using Weighted Majority Prediction Algorithms. Proceedings of the Fifteenth International Conference on Machine Learning, (ICML 1998), Madison, Wisconsin, USA, July 24-27, 1998, 395-403.
21. Pazzani, M. J., Billsus, D. Content-Based Recommenda-tion Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (Eds.), The Adaptive Web, Lecture Notes in Computer Science, 4321, Springer, Berlin-Heidelberg, 2007, 325-342. https://doi.org/10.1007/978-3-540-72079-9_10
22. Rapečka, A., Dzemyda, G. A New Recommendation Model for the User Clustering-Based Recommendation System. Information Technology and Control, 2015, 44(1), 54-63. https://doi.org/10.5755/j01.itc.44.1.5931
23. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L. BPR: Bayesian Personalized Ranking from Implicit Feedback. Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, (UAI 2009), Montreal, Quebec, Canada, June 18-21, 2009, 452-461.
24. Rezaee, M. R., Lelieveldt, B. P. F., Reiber, J. H. C. A New Cluster Validity Index for the Fuzzy C-Mean. Pattern Recognition Letters, 1998, 19(3-4), 237-246. https://doi.org/10.1016/S0167-8655(97)00168-2
25. Rodriguez, A., Laio, A. Clustering by Fast Search and Find of Density Peaks. Science, 2014, 344(6191), 1492-1496. https://doi.org/10.1126/science.1242072
26. Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., Monfardini, G. The Graph Neural Network Model. IEEE Transactions on Neural Networks, 2009, 20(1), 61-80. https://doi.org/10.1109/TNN.2008.2005605
27. Wang, J., Li, J., Li, J., Lai, W. Adaptive Fast Search Den-sity Peak Clustering Algorithm. Computer Engineering and Applications, 2019, 55(24), 122-127. https://doi.org/10.3778/j.issn.1002-8331.1901-0086

28. Wu, B., Lu, H., Jiang, H. Adaptive Density Peaks Clus-tering Algorithm. Journal of Computer Applications, 2020, 40(6), 1654-1661. https://doi.org/10.11772/j.issn.1001-9081.2019111881

29. Wu, L., Sun, P., Fu, Y., Hong, R., Wang, X., Wang, M. A Neural Influence Diffusion Model for Social Recommen-dation. Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Infor-mation, (SIGIR 2019), Paris, France, July 21-25, 2019, 235-244. https://doi.org/10.1145/3331184.3331214

30. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P. S. A Comprehensive Survey on Graph Neural Networks. IEEE Transactions on Neural Networks and Learning Systems, 2021, 32(1), 4-24. https://doi.org/10.1109/TNNLS.2020.2978386

31. Xie, J., Gao, H., Xie, W. K-nearest Neighbors Optimized Clustering Algorithm by Fast Search and Finding the Density Peaks of A Dataset. SCIENTIA SINICA Infor-mationis, 2016, 46(2), 258-280. https://doi.org/10.1360/N112015-00135-20

32. Yang, J., Yecies, B. Mining Chinese Social Media UGC: A Big-Data Framework for Analyzing Douban Movie Re-views. Journal of Big Data, 2016, 3, 1-23. https://doi.org/10.1186/s40537-015-0037-9

33. Zahir, A., Yuan, Y., Moniz, K. AgreeRelTrust-A Simple Implicit Trust Inference Model for Memory-Based Collab-orative Filtering Recommendation Systems. Electronics, 2019, 8(4), 427. https://doi.org/10.3390/electronics8040427

34. Zhang, S., Yao, L., Sun, A., Tay, Y. Deep Learning Based Recommender System: A Survey and New Perspectives. ACM Computing Surveys, 2019, 52(1), 1-38. https://doi.org/10.1145/3285029

35. Zhang, Z., Kudo, Y., Murai, T. Neighbor Selection for User-Based Collaborative Filtering Using Covering-Based Rough Sets. Annals of Operations Research, 2017, 256(2), 359-374. https://doi.org/10.1007/s10479-016-2367-1

36. Zheng, B., Hu, J. AANMF: Attribute-Aware Attentional Neural Matrix Factorization. Information Technology and Control, 2019, 48(4), 682-693. https://doi.org/10.5755/j01.itc.48.4.23149

37. Zhou, W., Cao, D., Xu, Y., Liu, B. A Survey of Recom-mendation Systems. Journal of Hebei University of Science and Technology, 2020, 41(1), 76-87. https://doi.org/10.7535/hbkd.2020yx01009