International Journal of Computational Intelligence Systems (2024) 17:204
https://doi.org/10.1007/s44196-024-00602-9

RESEARCH ARTICLE

A Fairness Group Recommendation Algorithm Based On User Activity

Junjie Jia[1] · **Fen Wang**[1] · **Huijuan Wang**[1] · **Shilong Liu**[1]

Received: 9 April 2024 / Accepted: 10 July 2024
© The Author(s) 2024

Abstract
In the process of group recommendation, due to the different preferences of group members, the recommendation results cannot meet the needs of all users. How to maximize the fairness of group recommendation is still a challenge. Therefore, this paper proposes a group recommendation algorithm based on user activity. Firstly, a group discovery algorithm based on item cluster preference was used to mine potential groups. Secondly, considering the dynamic change of activity, a sliding time window is designed to investigate the recent activity of each member in the group at the time of subgroup division, and the group is divided into active subgroup and inactive subgroup. Finally, the group recommendation list was generated by aggregating the subgroup preferences by average consensus. Experimental results on the public dataset show that compared with the AGREE algorithm, the recommendation accuracy and coverage of the proposed algorithm are improved by 2.1% and 2.9%, respectively. By focusing on the preference needs of inactive users, the proposed algorithm effectively improves the recommendation satisfaction and group fairness.

Keywords Group recommendation · Fairness · Activity · Attention mechanism

1 Introduction

With the rise of the Internet era, how to alleviate the problem of information overload and quickly and accurately obtain the useful content in the data is an urgent problem to be solved by the recommendation system [1]. By means of collaborative filtering [2] and matrix decomposition [3], the traditional personalized recommendation system has achieved great success in prediction ratings and recommendations for individual users, and is widely used in e-commerce [4], music, news and other fields. With the continuous innovation of transportation and media technology, people are more and more inclined to communicate with each other in the form of groups, such as travel, dinner and so on. However, there are bound to be differences and conflicts in group behavior. How to eliminate the unfairness caused by the personalized differences among users in a group and how to recommend groups according to the preferences of members in a group has become an important issue in the recommendation system. Therefore, group recommendation comes into being [5].

Group recommendation system is a technology applied to social platforms or online communities [6]. It aims to gather multiple similar users to form a group, integrate the consensus preferences of different members in the group and alleviate the personal preference conflicts to the greatest extent, so that the recommendation results can meet the needs of all/most members. At present, the research on group recommendation mainly focuses on preference fusion, as shown in Table 1. Although these studies have promoted the process of preference fusion to different degrees from different perspectives, there are still problems that the fusion strategy only reflects the preferences of the main users (such as the most respected strategy, the most happy strategy, the least pain strategy, etc.) or the majority of users (such as the average strategy, etc.) [7]. These fusion strategies do not consider the activity difference between different members of the group. At the same time, some progress has been made in the research of group behavior in social psychology. According to the symbolic interaction theory of Blumer [8], member behavior will continue to reflect and infect in the group, and finally achieve the consistency of group behavior. Back [9] believes that the formation of group behavior depends on

✉ Fen Wang
2314136706@qq.com

[1] School of Computer Science and Engineering, Northwest Normal University, Lanzhou, Gansu, China

Published online: 05 August 2024

Table 1 Commonly used fusion strategies in group recommendation algorithms

Fusion strategy	Description	There are problems
Mean strategy [12]	The average rating of the group members is the group score	Ignoring differences in members of the group
Minimum pain strategy [13]	The lowest rating among the group members is selected as the group score	Ignoring positive emotions leads to the marginalization of opinions about certain members
The happiest Strategy [14]	The highest rating among group members is selected as the group score	The participation of group members is uneven, and the needs of different members are ignored
The most admired strategy [15]	The core member representative score was selected as the group score	Focus only on the leaders in the group
Fair strategy [16]	The member rating is selected as the group rating in turn, and the recommended item is from the individual highest rating item. When the item rating is the same, the opinions of other group members are considered	The needs and interests of inactive members are ignored, resulting in a reduced sense of participation
Multiplication strategy [17]	The rating of the item is expressed as the product of the ratings of all members on the item	Overemphasis on the views of some members causes the views of others to be ignored or devalued
Borda counting strategy [18]	Each item is given a score based on its position in the personal preference list, and the scores are summed as the group score	Does not adequately reflect the actual level of participation of members

the mutual stimulation of participants, and special members such as leaders or active members of the group play a leading role in group consensus behavior. However, Page [10] believes that we should not only focus on special members and ignore the opinions of some inactive members, because the dissent within the group will enable people to obtain more information, think about problems in new ways, and often make better decisions. Therefore, as a group behavior, group recommendation should not only focus on the core users or some active users, but also consider the contribution of inactive users to the group preference to generate group recommendation decisions with maximum satisfaction of all users. This is of great significance for coordinating the fairness of users in the group and reducing the herd behavior and social loathing of individual users [11].

In the real recommendation scenario, the activity a_u of user is generally measured by the interaction frequency (click, download, etc.) between user u and all items. To analyze the influence of user activity on item recommendation, this paper observed the relationship between user activity and average popularity of items on the MovieLens-100 K dataset (Fig. 1).

The popularity of an item v is generally expressed in terms of the item's audience size n_v, and the number of users interacting with the item. Here, to illustrate the relationship between popular or non-popular movies and users, the corresponding popularity $n_{v_1}, n_{v_2}, \ldots, n_{v_k}$ of the historical watched movies $v_u = [v_1, v_2, \ldots, v_k]$ of user u is averaged, and the user's interest tendency is expressed in the form of average popularity $POP_u = \frac{1}{k}\sum_{i=1}^{k} n_{v_i}$. As can be seen from Fig. 1, by plotting the relationship between user activity

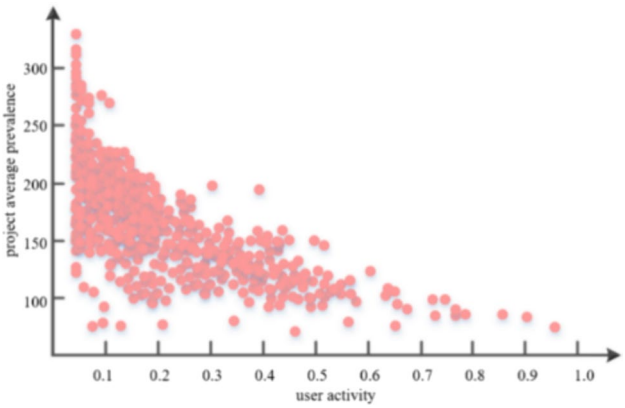


Fig. 1 Relationship between user activity and item popularity

and the average popularity of items, it can be observed that users with lower activity are more likely to choose items with higher popularity, and users with higher activity are more likely to explore unpopular (low popularity) items. Users with lower activity are more likely to choose items with higher popularity. This may be because these items are actively recommended and discussed by other users in the group, thus attracting the attention of users with lower activity, who want to select some items from the group that are recognized and recommended by the majority to avoid in-depth research and screening of a large number of items. On the contrary, users with higher activity are more likely to choose items with lower popularity. This may be because the highly active users have some knowledge of the popular items in the group, and they are more willing to try and find some less popular but outstanding quality items, so as

to satisfy their curiosity and pursue personalized viewing experience. Therefore, modeling the recommendation task from the perspective of activity is beneficial to broaden the recommendation scope of items.

To verify the role of activity opinion in the group recommendation task, this paper generates multiple groups according to the commonly used k-means method and analyzes all the data and the distribution of users under different activity in several groups (Fig. 2).

As can be seen from Fig. 2, among all users, the activity difference among various users is obvious. From the observed data, the activity of some users is less than 0.1, and that of some users is more than 0.9 or even close to 1. As the interaction between users and movies becomes more frequent (activity increases), the activity of users follows the long tail distribution, indicating that a few users have very high or very low activity, while most users mainly gather in a low activity range. Further, it can be found that in different groups, the corresponding number of users under different activities is generally low, which indicates that the activity of different users in the group is different, that is, the user activity in the group is diversified. It can be seen that the common fusion strategies that only focus on some users' preferences (Table 1) are not conducive to meeting the preference needs of other users. Therefore, it is realistic and logical to achieve recommendation tasks by comprehensively focusing on the preference needs of active and inactive users.

The research in this paper focuses on group recommendations rather than on social networking platforms geared toward large-scale user participation. This means that the research goal of this paper is not concerned with social structures with large user groups and complexity, such as Facebook or Twitter. Instead, this paper focuses on smaller, more specialized social groups or domain-specific professional communities. These communities are likely to be made up of people with in-depth knowledge of a particular field, who communicate with each other more intensively and professionally and are more inclined to share content on specific topics. Although the focus of this paper is not on large social networking platforms, the research features and strengths of this paper can be more clearly demonstrated by showing the effectiveness of our approach to personalized recommendation in the context of small-scale social groups. By choosing small-scale social groups as our research object, we can focus our efforts on how to satisfy the needs of these specific groups through personalized recommendation methods. The goal of this paper is to develop recommendation algorithms that are more accurate and adaptable to these particular environments to improve user satisfaction and engagement. This is different from the research on recommender systems on large-scale social networking platforms, but is equally important for research and practical applications.

Due to the different levels of activity of users, there are differences in the contribution to the group. In most cases, group preference as consensus wisdom can only meet the needs of some more active users, and it is often difficult to

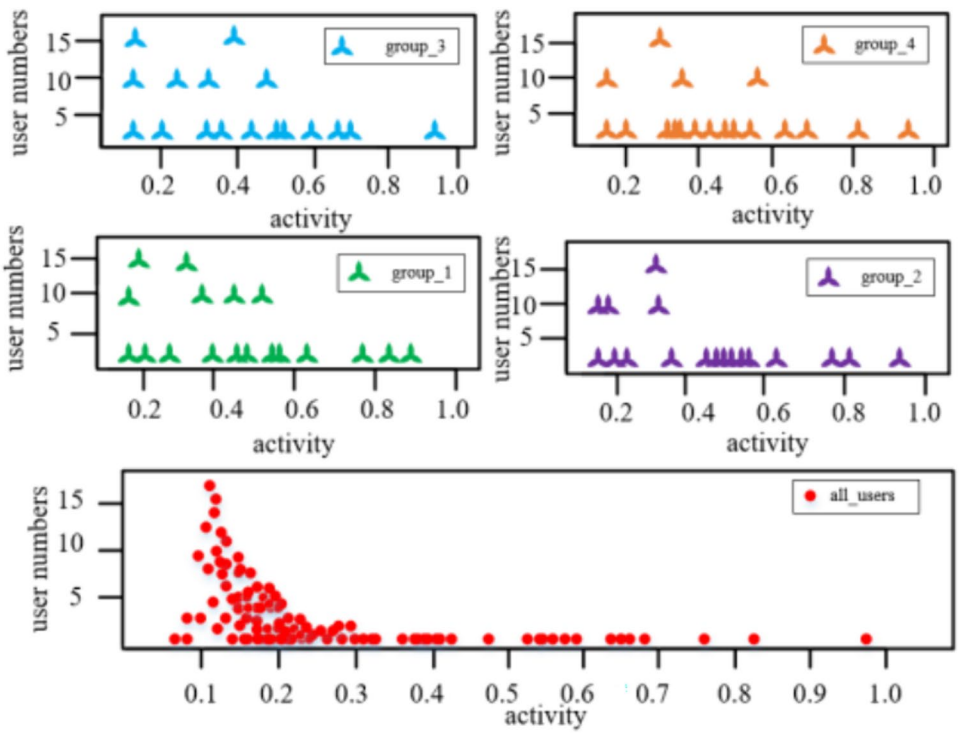


Fig. 2 Relationship between number of users and activeness

meet the recommendation needs of low or inactive users [19]. To achieve the fairness goal of group recommendation, users are divided into active users and inactive users, and the preferences of two subgroups are obtained, respectively, to form the preferences of the whole group. This approach has two advantages. First, although the inactive group has a low contribution to the group consensus, its needs as group members cannot be ignored or suppressed, which is detrimental to the fairness of recommendation and the ecological and healthy development of the recommendation system. Second, the preferences of the two subgroups, respectively, have a high cohesion group consensus, which is more conducive to improving the cohesion of the group preference consensus.

According to the above theory, it is necessary to comprehensively consider the influence of active and inactive users on group preference. From the perspective of practical applications, the following three key issues need to be considered: first, the representation learning of higher-order implicit relationships between users. As a special group, in the absence of external auxiliary information (social relationship), there is a certain implicit relationship between individuals, which mutually influence their interests and decision-making behaviors. For example, purchasing the same goods reflects the similarity of interests among users. Therefore, acquiring a generalized representation of implicit influence can reflect the contribution of users to the formation of group consensus. Additionally, the implicit relationship between users also impacts their interest in items. Therefore, the user–item and user–user correlations should be comprehensively considered in the user activity degree. Thirdly, focusing on active and inactive user subgroups can expand the breadth and depth of group recommendation decisions. How to integrate the consensus preferences of the two subgroups to form a group recommendation list is also a key factor affecting the fairness of group recommendation.

In this paper, we propose a fair group recommendation algorithm based on user activity level (FUA) to overcome this potential problem and improve the quality of recommendation. The existing group recommendation algorithms directly model the group preferences according to the preferences of the members. This modeling method will make the recommendation system pay too much attention to the preferences of the active members and ignore the influence of the inactive members on the recommendation results. Therefore, this paper learns the implicit interaction embeddings of users and items according to the user–item bipartite graph, detects the activity of group members through the attention network, and establishes two subgroups, active and inactive, and uses the self-attention mechanism to capture the subgroup preferences to generate a two-group recommendation list. The average consensus aggregation Top-n strategy is proposed to merge the two-group recommendation lists to generate a group recommendation list. Group recommendation algorithms usually use consensus preferences instead of individual preferences to make group recommendations. However, there may be large differences in the preferences of group members. As a result, in the process of group recommendation, the results of group recommendation may not be satisfied by each group member, and even some user preferences that are significantly different from the group's preferences cannot be paid attention to and are ignored, which makes the fairness of group members not be guaranteed. The contribution of users with large preference differences from other members of the group to the recommendation of products should not be ignored. Therefore, the activity of members in the group is detected, and the recent activity of each member is analyzed before preference fusion. According to the activity, the group is divided into groups, so as to coordinate the fairness of users in the group, reduce the unreasonable feedback of individual users, and help to improve the quality of the group recommendation list and maintain the long-term development of the group. Compared with other aggregation strategies, the proposed algorithm considers the contribution of inactive members to group recommendation, which can effectively improve the satisfaction of group members, especially in the aspect of member fairness.

The main contributions of this paper are as follows:

1. This paper proposes a high-order representation model based on user–item bipartite graph. Graph neural network is used to learn the embedded representation of high-order implicit interactions between users and items, which contains more information.
2. This paper proposes a multi-layer attention two-group division method based on activity, according to the attention network, the user activity is calculated, and the group is divided into two subgroups: active and inactive. The self-attention mechanism is used to capture the subgroup preference, which helps to focus on the contribution of the inactive subgroup users to the group recommendation results.
3. This paper proposes a group recommendation algorithm with enhanced fairness. It generates two groups of recommendation lists by subgroup preference and merges them to generate group recommendation lists according to the average consensus aggregation Top-n strategy, which effectively improves the fairness of group recommendation.

The remaining part of this paper is organized as follows: Section II describes the related work of this paper. Section 3

focuses on the algorithm, including the theoretical basis and implementation steps. In Sect. 4, the performance of the proposed algorithm is verified and analyzed through experiments. Section 5 concludes the work and gives the future work.

2 Related Work

Group recommendation recommends appropriate items or resources to a group through group activities. It extends the recommendation object from a single user to a group [20, 21] and provides strong suggestions for the group decision-making process to meet the needs of users in group activities [22]. From a technical point of view, existing group recommendation methods can be divided into two categories: memory-based methods and model-based methods. Memory-based methods adopt predefined strategies to aggregate the preferences of members in a group, which can be further divided into model fusion methods and recommendation fusion methods [5]. The model fusion method fuses the preferences of group members before recommendation. It fuses the individual preferences of group members to obtain the group preference model and then makes personalized recommendation according to the group preference model to generate the group recommendation list [23]. In contrast, the recommendation fusion method first generates a personalized recommendation list for each member of the group, and then fuses the personalized recommendation lists of members to obtain the final group recommendation list [7]. However, due to the different activity of each member in the group, their contribution to the group-recommended resources is also different, so the memory-based method cannot simulate the group decision-making process well.

Different from memory-based methods, model-based methods try to model the decision-making process of the group, which is used to integrate the preferences of different group members for items. Literature [24] proposed the personal influence topic model by analyzing the decision-making process of groups, which assumed that users and groups obeyed the Dirichlet distribution of topics and considered the personal preferences, personal influence and social network information of group members. The influential users could represent the group decision-making, but this model did not consider the influence of users in different models. Literature [25] proposed COM (consensus model) probability model to simulate the generation process of group activities and generate recommendation results for the group. However, model-based methods may rely on general group recommendation policies, and the fairness of group members is difficult to guarantee.

Literature [26] gave the definition of fairness, and considered the correlation between group members and recommended items in a rank-sensitive way, so as to find a group recommendation list that met the definition of fairness. Literature [27] proposed AGREE (Attentive Group Recommendation) algorithm, which solves the preference aggregation problem by learning aggregation strategies from data and uses neural network to learn the interaction process to improve the accuracy of group recommendation. But this method only considers the complex decision-making process of the group. The inter-user influence is not considered, and the fairness of group members is not guaranteed. Literature [28] proposed a group recommendation algorithm SIG (Social Influence-based Group) based on social influence, which can learn user characteristics and social influence from data uniformly. However, this method does not take into account that some members may only be influential in certain decisions rather than in the whole group. Literature [29] proposed FAI algorithm, which successively obtains the favorite items of each member of the group and adds them to the group recommendation list. Although this method takes into account the preferences of all members in the group, there are numerous recommendation results available, which hinder selection for users and increase the complexity of the algorithm. Literature [5] uses non-negative matrix factorization to analyze the professional degree of members, only focuses on the influence of professional members in the group on the fusion result, ignores the influence of inactive members that reduces the influence of inactive members on the fusion result, and does not pay attention to the fairness of group recommendation.

Graph neural networks have been successfully implemented in many fields [30], such as chemistry [31], physics [32], commonsense reasoning [33], natural language processing [34], social networks [35], traffic flow prediction [36, 37], and medicine [38–40]. Literature [41] proposed a random walk graph convolutional network (GCN)-PinSage. pinSage is a highly scalable GCN algorithm capable of learning node embeddings in a web-scale graph containing billions of objects. Literature [42] proposes a new framework for recommender systems based on graph convolutional neural networks (GCNN) called multi-graph convolutional collaborative filtering (Multi-GCCF). Experiments on four datasets with more than one million user–item interactions verified that MultiGCCF outperformed the state-of-the-art models at the time. Literature [43] proposes the NIA–GCN framework, which explicitly models relationship information between neighboring nodes and exploits the heterogeneity of the user–item bipartite graph. Significant improvements

were observed on a large-scale commercial dataset with more than 25 million user–item interactions, demonstrating the practical potential of graph neural networks applied to large-scale commercial recommender systems. Literature [44] proposes a flexible and highly scalable side-information (SI)-enhanced skip-gram (SISG) framework, currently deployed on Taobao. The framework allows the computation of item and side information embedding vectors over product sets of billions of size. Literature [45] proposes a graph neural network-based solution to reconstruct loose item sequences into tight item–item interest graphs. Experiments are conducted on two large public datasets, Taobao and Kuaishou, to verify the effectiveness of graph neural networks in the recommendation domain. Literature [46] proposes graph convolution machine (GCM), an end-to-end framework consisting of three components: an encoder, a graph convolution (GC) layer and a decoder. Experiments were conducted on three real datasets from Yelp and Amazon to validate the effectiveness of the GCM and the advantages and efficiency of using graph neural networks in recommender systems. Literature [47] proposes a session-based social recommendation model, GNNRec, which uses graph neural networks to efficiently model users' interests. Experiments were conducted on two large datasets, Douban and Epinions, and the experimental results showed that the performance of the GNNRec model was significantly better than the mainstream recommendation models at the time. Literature [48] proposes a novel session recommendation method, SR-HetGNN, which uses heterogeneous graph neural networks (HetGNNs) to learn session embeddings and capture specific preferences of anonymous users. Through extensive experiments on two large datasets, Diginetica and Tmall, SR-HetGNN is shown to outperform existing state-of-the-art session-based recommendation methods.

Inspired by the above research, this paper uses the attention mechanism to analyze the member activity. By dividing the group into active and inactive subgroups, this paper tries to pay attention to the influence of inactive members on the recommendation results, to establish a group preference model with higher accuracy and improve the fairness of group recommendation.

3 Model

3.1 Problem Formulation

Aiming at the fairness problem that the needs of a few cold users are not paid attention to in the group recommendation system, this paper uses the attention mechanism to detect the user activity, divides the group into active and inactive subgroups, and obtains the preferences and recommendation proportion of each subgroup. The group decision-making process in real scenarios is simulated, and the group recommendation list is obtained. By paying attention to the preferences of inactive users, the algorithm improves the satisfaction of the whole group users and realizes the fairness of group recommendation. Figure 3 shows the overall framework of the proposed algorithm. In the first stage, this paper uses a group discovery algorithm based on item cluster preferences to cluster users and discover potential groups; in the second stage, within

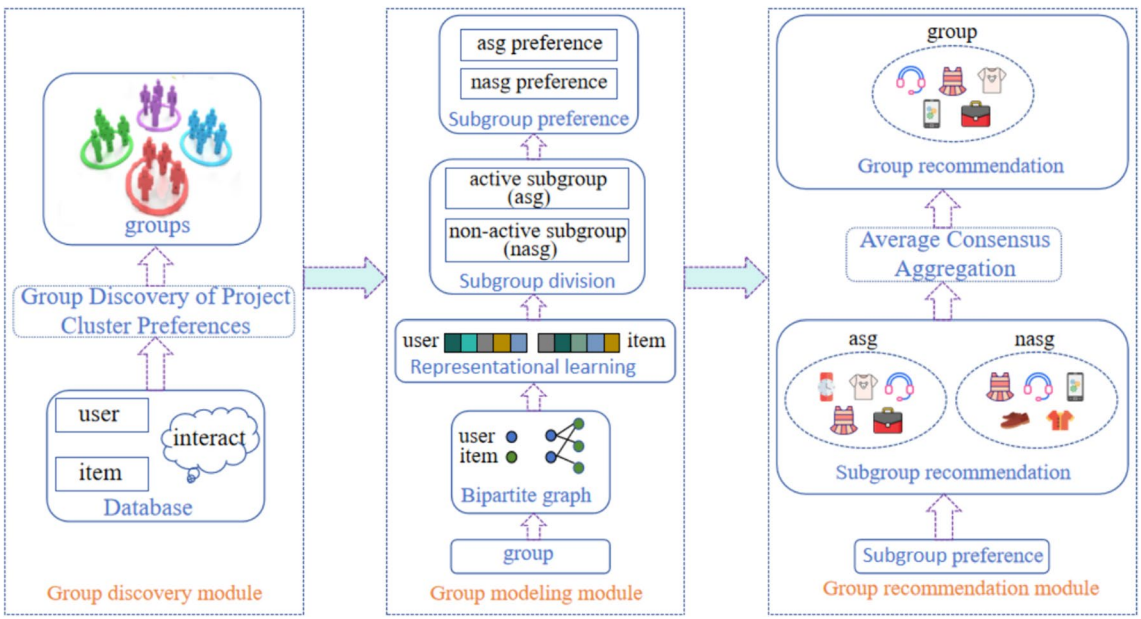


Fig. 3 Overall framework diagram of the FUA algorithm

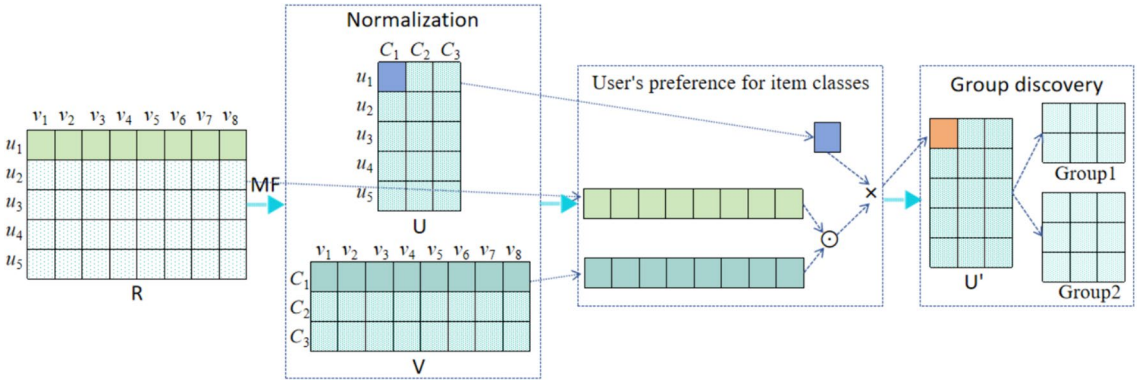


Fig. 4 Group discovery of project cluster preferences (GDPCP)

the observed time window, firstly, the user–item bipartite graph and attention are used for characterization learning, secondly divide the group into subgroups according to the level of user activity in the group, and finally use the NCF model to obtain the preferences of the subgroups; in the third stage, the candidate recommendation lists of the two subgroups based on the subgroup preferences are obtained, and then the average consensus preference strategy proposed in this article fuses two candidate recommendation lists to obtain the group's recommendation list.

According to Fig. 3, the proposed algorithm mainly includes the following three modules:

1. Group discovery: The group discovery module is implemented based on NMF dimensionality reduction and K-means clustering, which is mainly used to find more similar groups inside, which is the potential basis of this paper.
2. Group modeling: The group modeling module first divides the group into active and inactive subgroups according to the member activity obtained by the attention network, then uses the self-attention network to obtain the representation of each subgroup, and further uses the NCF model to obtain the preferences of two subgroups, which is the core content of this paper.
3. Group recommendation: The group recommendation module uses Top-n recommendation method on the established model to generate a recommendation list for the group, which is the main purpose of this paper.

The above is the main content of this paper, and in the later part, this section will be expanded in turn according to the above modules.

3.2 Group Discover

K-means clustering algorithm is considered as one of the most commonly used clustering algorithms due to its advantages of simple implementation and fast convergence speed. In Ref. [5], a group discovery algorithm was proposed, which first used non-negative matrix factorization NMF to iteratively decompose the user–item rating matrix $R_{m\times n} = U_{m\times k} \times V_{k\times n}$ to generate the predicted rating, and then input the final decomposed user matrix U into the K-means clustering algorithm to mine the potential groups. However, ignoring the influence of item matrix V on user matrix$U$, the latent factor k directly determines the final clustering effect. Based on [25], this paper proposes a new group discovery algorithm based on project cluster preferences (GDPCP), which consists of three steps. 1) Normalization operation. The ith user vector in the user matrix after NMF decomposition is normalized to $u_i = \{h_{ij}|j \in [1,k]\}$, h_{ij} is used as the preference weight of user u_i to the jth class C_j in k potential item clusters, and the normalized user matrix is U. The ith item cluster C_i in the item matrix is normalized to $C_i = \{v_{ij}|j \in [1,n]\}$, v_{ij} represents the jth component value of the item cluster C_i, indicating the weight of all items in C_i, and the normalized item matrix is V. 2) Calculate user preferences for item clusters. If the scoring vector of user u_i in the scoring matrix R is R_i, then the preference of user u_i for project cluster C_j is $r_{ij} = h_{ij} \sum R_i \odot C_j$, $\sum R_j \odot C_j$ calculates the weighted sum of all scores of user u_i with C_j as the weight, which represents the score of user u_i to C_j, and $\odot$ is the element-level dot product operation. 3) Group discovery. The updated user matrix U' is input into the K-means clustering algorithm to mine the potential groups. The NMFS algorithm process is shown in Fig. 4. The algorithm uses the item cluster C in the item normalization matrix V after MF decomposition to weighted sum the user rating matrix R [29], calculates the user rating matrix for the item cluster C based on the user preferences in the user normalization matrix U, and applies the K-means algorithm to cluster the matrix according

to the user to identify the group. While considering the importance of each item in the item cluster, the algorithm reduced the sparsity of the user rating matrix to avoid the impact of "dimensional disaster". Users with similar interests were clustered into the same group, and users with different interests were located in different groups. In this paper, the cosine distance is used to measure the similarity between users. The cosine distance is calculated by the following Eq. (1):

$$dist\left(\vec{A},\vec{B}\right) = 1 - \cos\theta = 1 - \frac{\vec{A}\cdot\vec{B}}{\left|\vec{A}\right|\cdot\left|\vec{B}\right|}, \tag{1}$$

where $\vec{A}, \vec{B}$ denote two vectors, θ denotes the angle between vectors, and $\left|\vec{A}\right|$ denotes the modulus of vectors $\vec{A}$.

The group discovery algorithm is described as follows:

Algorithm 1 The group discovery algorithm GDPCP

Input: user-item rating matrix $R^{m\times n}$, number of latent factors k, number of groups Q.
Output: Q groups.
Steps 1 $(U^{m\times k},V^{k\times n}) \leftarrow R^{m\times n} \leftarrow$ NMF;
Steps 2 $U^{m\times k},V^{k\times n} \leftarrow U^{m\times k},V^{k\times n} \leftarrow$Normalization
Steps 3 $U'^{m\times k} \leftarrow R^{m\times n},U^{m\times k},V^{k\times n} \leftarrow$Weighted fusion
Steps 4 $U'^{m\times k} \leftarrow$ K-means;
Steps 5 Return Q groups.

3.3 Group Modeling

In the existing group recommendation systems, the recommended items cannot reflect the preferences of all members in the group, and generally only consider the needs of the majority of active users while ignoring the needs of a few cold users [19]. How to improve the consensus preference needs of the group while improving the individual satisfaction of the group members is an urgent problem to be solved.

3.3.1 User–Item Bipartite Graph Representation Learning

In recommender systems, the most intuitive expression of whether a user is active or not is the number of their historical ratings. However, for group recommender systems, it is impractical to analyze the activity of group members over the entire time series when examining activity, because users who were active before are not necessarily active now, and users who were silent before are not necessarily silent now. In short, it is the case that user behaviors that are too old to influence the current activity of a user are merely noise to the current recommendation. In addition, recommendation generation can be affected by many unforeseen factors. These factors may lead to significant changes in user activity after a specific moment. Therefore, in this paper, the examination of member's activity is set to be within a period of time before the moment of subgroup division, based on which a sliding time window is designed, as shown in Fig. 5:

In Fig. 5, the maximum and minimum time stamps in the behavioral data are first observed, denoted as t_{max} and t_{min}, respectively. t_{max} denotes the time closest to the current time, while t_{min} denotes the time furthest from the current time. The time interval $|t_{max} - t_{min}|$ is then divided into many small segments, each representing a week (604,800 s), called short windows, denoted by Δt. The η short windows form a long window, denoted as T ($T = \eta\Delta t$), and the long window T is set to slide forward with time. Finally, assuming that subgroups are divided at t_{sub} time, to analyze the activity of each member in the recent time period, it is sufficient to slide the long window to t_{sub} time and examine the activity of the group members in that window.

Since the user's interaction items within the current time window t can reflect the user's current preferences to a large extent, this makes it possible to measure the preference similarity between users. Inspired by the graph embedding approach in the literature [49], based on the lightweight graph convolution model [50], we propagate embeddings in user–item bipartite graphs, obtain embedding vectors of users and items, and use the global sense field of the self-attention mechanism [51, 52] to generate more valuable embedding representations by capturing the correlations between users or items and focusing attention on important features.

Before dividing subgroups based on user activity, users and items need to be characterized. To enrich the connotation of the characteristics of users and items, during the learning process of user–item bipartite graph representation, for the same user, its representation should incorporate the preference characteristics presented by all the items in its current time window t. Similarly, for the same item, its representation should reflect the combined interest of all the

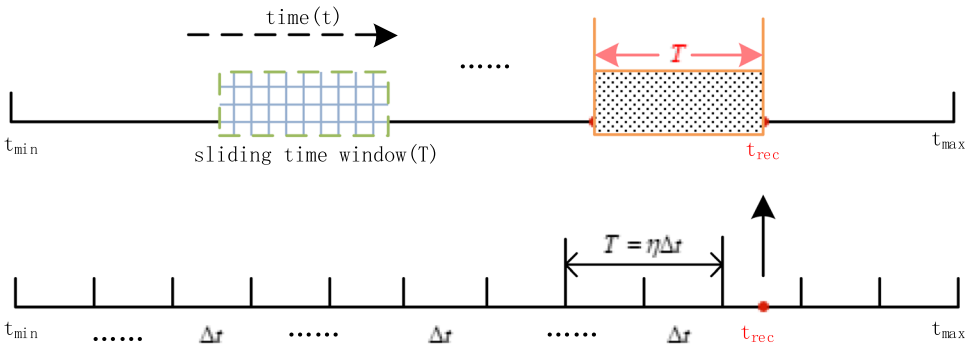


Fig. 5 Time sliding window

users in the item. This interactive synergy can propagate through multiple layers during representation learning to capture higher-order neighbor influences of users and items. Specifically, the representation learning process includes four main operations: 1) interactive embedding, which maps user or item features and interactive propagation information into embedding vectors after fusion; 2) interactive aggregation, which generalizes the embedding representations of users or items by aggregating the information embeddings propagated from their neighbor nodes; 3) high-order propagation, to capture the interactive synergy effect in high-order connectivity, multiple propagation layer embeddings are integrated to generate content-rich feature representations; 4) attention representation, which uses the self-attention mechanism to obtain attention representation by dynamically learning the dependencies between users or items.

3.3.1.1 Interactive Embedding Assume a group $G = \{u_1, u_2, \ldots, u_k\}$ with k users in time window t, and v a collection of items$V = \{v_1, v_2, \ldots, v_m\}$. For a user–item pair (u, v) in the graph, in time window t, let the initial embedding vectors of the user u and the item v be denoted by $e_u^{(0)}(t), e_v^{(0)}(\mathrm{t}) \in R^d$, respectively, where d denotes the dimension of the embedding vector. Then, the interaction embedding $e_{v\to u}^{(i)}(t)$ propagated from item v to user u and the interaction embedding $e_{u\to v}^{(i)}(t)$ propagated from user u to item v at the ith order are defined as:

$$e_{v\to u}^{(i)}(t) = B\left(w_1^{(i)} e_v^{(i-1)}(t) + w_2^{(i)}\left(e_v^{(i-1)}(t) \odot e_u^{(i-1)}(t)\right)\right), \quad (2)$$

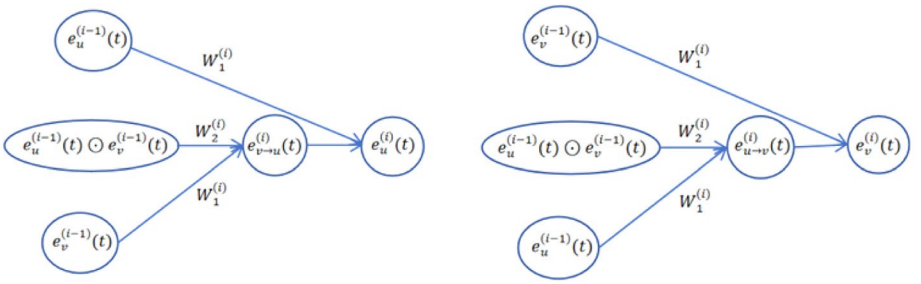
$$e_{u\to v}^{(i)}(t) = B\left(w_1^{(i)} e_u^{(i-1)}(t) + w_2^{(i)}\left(e_u^{(i-1)}(t) \odot e_v^{(i-1)}(t)\right)\right), \quad (3)$$

where $\odot$ is the element-wise product; $w_1^{(i)}$ and $w_2^{(i)}$ are the network parameters of the ith propagation layer of the model, and the attenuation factor $B = 1/\sqrt{|N_u||N_v|}$ follows the GCN normalized design [49].N_u denotes the set of rank ith adjacent item nodes that directly interact with user u, and N_v denotes the set of rank ith user nodes that are adjacent to item$v$. $|N_u|$ and $|N_v|$ denote the number of N_u and N_v, respectively.

3.3.1.2 Interaction Aggregation In this phase, the propagation information of users u or items v with the ith order neighbor nodes is aggregated to generalize the embeddings of u and v to represent $e_u^{(i)}(\mathrm{t}), e_v^{(i)}(\mathrm{t})$. The aggregation function is defined as follows.

$$e_u^{(i)}(\mathrm{t}) = \sigma\left(w_1^{(i)} e_u^{(i-1)}(\mathrm{t}) + \sum_{v\in N_u} e_{v\to u}^{(i)}(\mathrm{t})\right), \quad (4)$$

$$e_v^{(i)}(\mathrm{t}) = \sigma\left(w_1^{(i)} e_v^{(i-1)}(\mathrm{t}) + \sum_{v\in N_v} e_{u\to v}^{(i)}(\mathrm{t})\right). \quad (5)$$



(a) Embedding and Aggregation of $e_u^{(i)}(\mathrm{t})(u^i(\mathrm{t})EA)$ (b) Embedding and Aggregation of $e_v^{(i)}(\mathrm{t})(v^i(\mathrm{t})EA)$

Fig. 6 The ith embedding and aggregation layer (iEA). **a** Embedding and aggregation of $e_u^{(i)}(\mathrm{t})\left(u^i(\mathrm{t})EA\right)$. **b** Embedding and aggregation of $e_v^{(i)}(\mathrm{t})(v^i(\mathrm{t})EA)$

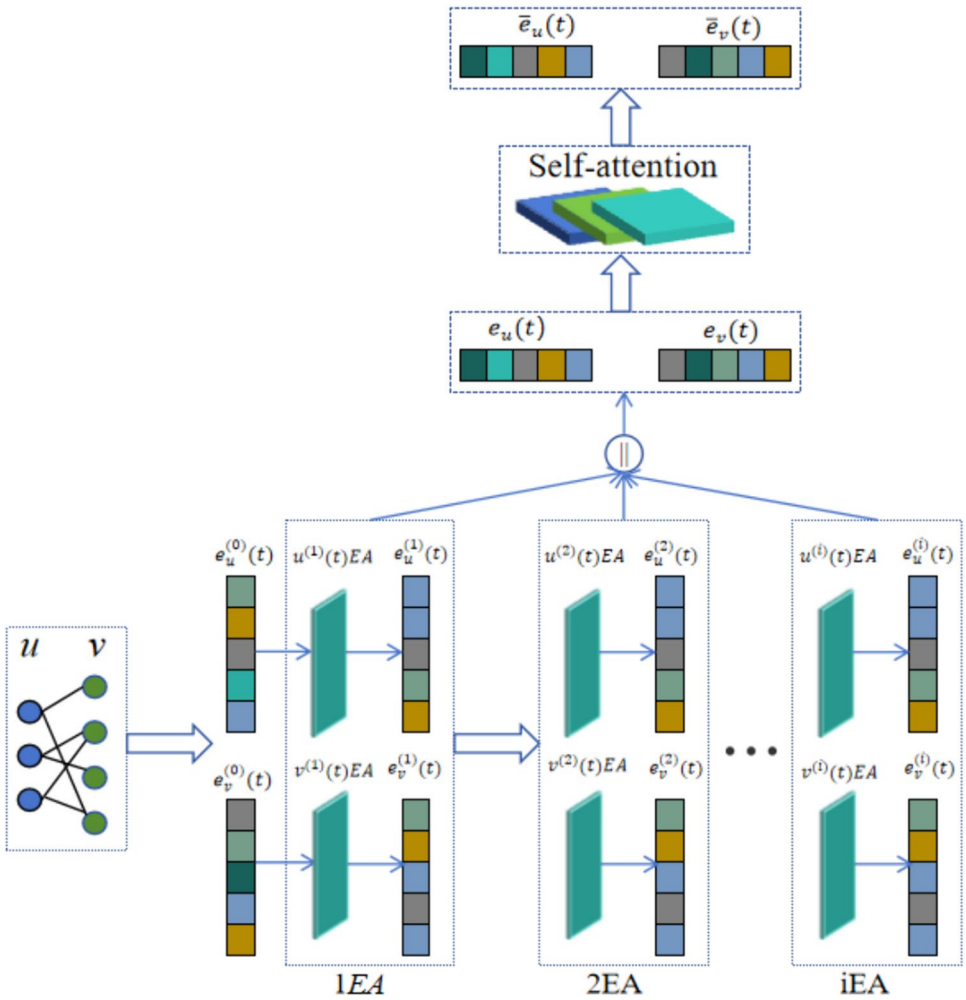


Fig. 7 Higher-order propagation and attention representation

Here, σ is the activation function. The ith order feature embeddings of users and items incorporate the combined effects of direct and indirect interactions between users and items, and this higher-order connection relationship is crucial to capture the synergistic effects between users and items. The embedding and polymerization process(iEA) of $e_u^{(i)}(t)$ and $e_v^{(i)}(t)$ is shown in Fig. 6.

3.3.1.3 Higher-Order Propagation To explore higher-order connectivity synergies within the current time window t, users and projects need to be embedded by fusing multiple interaction propagation layers (Fig. 7). After the ith layers of propagation, the multiple embeddings of user u are denoted as $\left\{e_u^{(1)}(t), e_u^{(2)}(t), \ldots, e_u^{(i)}(t)\right\}$ and the multiple embeddings of item v are denoted as $\left\{e_v^{(1)}(t), e_v^{(2)}(t), \ldots, e_v^{(i)}(t)\right\}$. Since the embeddings captured from different propagation layers cover different interaction attention contents and ranges, the multi-layer embeddings are fused to generate representations $e_u(t)$ and $e_v(t)$ with richer connotation, as shown below, where $\|$ is the join operation. θ_u and θ_v are the network weight parameters, respectively.

$$e_u(t) = f_{\theta_u}\left(e_u^{(1)}(t) \| \ldots \| e_u^{(i)}(t)\right), \tag{6}$$

$$e_v(t) = f_{\theta_v}\left(e_v^{(1)}(t) \| \ldots \| e_v^{(i)}(t)\right). \tag{7}$$

3.3.1.4 Attention Representation For the group user sequence, the self-attention mechanism uses linear transformation to map it into the representation of query (Q), key (K) and value(V), and dynamically learns the similarity between each user (item) and other users (items) by calculating the similarity between the three dependencies. The embedded feature matrix $E_u(\mathrm{t}) = \left\{e_u(\mathrm{t}) | u \in G\right\}\left(E_u(\mathrm{t}) \in R^{k \times d}\right)$ of the group users is used as input, after parameters $W = (WQ, WK, WV)$ lin-

ear mapping for three matrices $(Q, K, V) = WE_u(t)$, between the Q and K^T matrix multiplication, and the weight of V is obtained by the softmax layer, that is, attention matrix Y, and the matrix multiplication between Y and V. An element-wise matrix summation operation is performed on the multiplication result and the embedding matrix E_u to obtain the attention feature representation $\overline{E}_u(t)$ of the final $E_u(t)$. The self-attention mechanism filters out less informative interactions by focusing on the interaction influence with the highest weight to avoid overfitting. The user self-attention feature representation model is as follows:

$$\begin{cases} (Q, K, V) = WE_u(t) \\ Y = softmax\left(QK^T\right) \\ \overline{E}_u(t) = \alpha YV + E_u(t) \end{cases}. \tag{8}$$

Note that before performing the summation of $\overline{E}_u(t)$, the previous term needs to be reshaped to ensure that it is the same as $E_u(t)$. The default value of the proportion parameter α is 0, and more weights can be assigned to YV through continuous learning. Similarly, for the embedded feature matrices $E_v(t) = \left\{e_v(t) | v \in V\right\}$ and $E_v(t) \in R^{m \times d}$ of items, it can be obtained through the self-attention mechanism the attention feature representation $\overline{E}_v(t)$ of $E_v(t)$ as shown in Fig. 7.

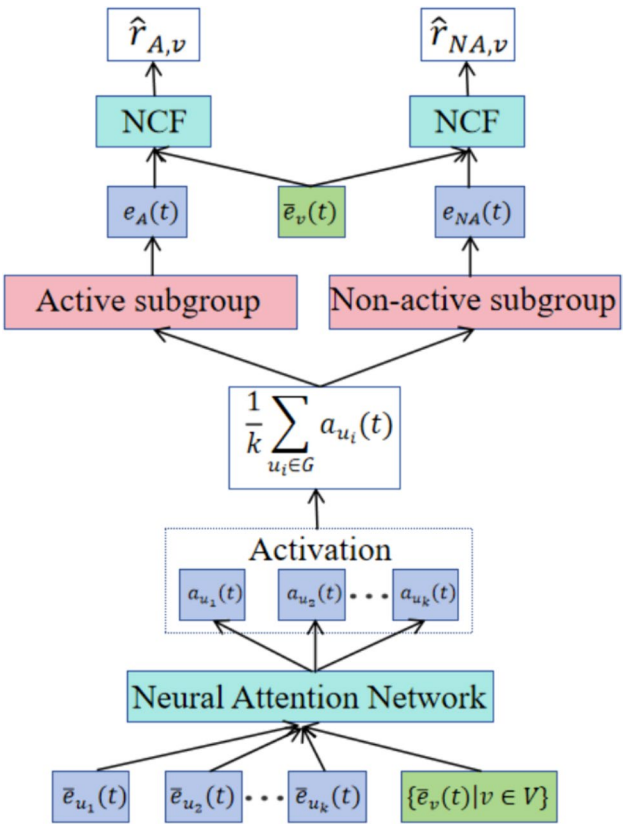


Fig. 8 Subgroup division and preferences

3.3.2 Subgroup Partitioning

Compared with users with active project interaction behaviors in a group, users with less interaction have less impact on the group recommendation task, and their needs are often ignored by the recommendation system [19]. In real scenarios, the more active individuals are, the greater contribution their preference information makes to the overall consensus preference, which makes the wisdom of inactive individuals be annihilated by the wisdom of active individuals in the process of converging individual wisdom to group wisdom. To achieve the fairness of group recommendation, the satisfaction of group members is maximized by comprehensively paying attention to the needs of active and inactive users. In this paper, we use the attention network [53, 54] to capture this user activity reflected from the interaction items, set the grouping threshold (the selection rules of the threshold are explained at the end of this section), divide the group into active and inactive subgroups, and generate subgroup embeddings based on the activity weights, and obtain the subgroup preferences after the neural collaborative filtering network NCF (neural collaborative filtering) [55–57] to obtain subgroup preferences. The subgroup division module is shown in Fig. 8.

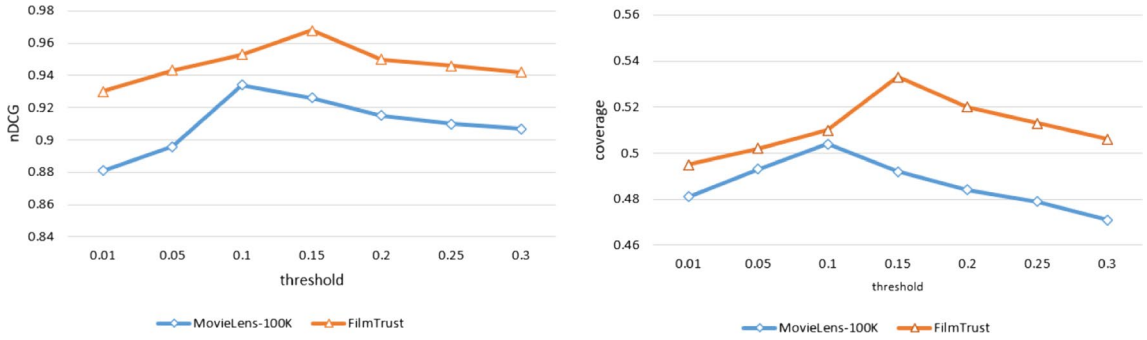
Firstly, the value of the influence of each interaction item on the user is obtained within the current time window t. The more items interacted with, the more are the items the user is influenced by and the more active the user is. According to the user and item embedding matrices $\overline{E}_u(t)$ and $\overline{E}_v(t)$, suppose that user u has interaction with item $v \in V$ in the current time window t, and let the value of the effect of v on u in dividing the subgroup moment t_{sub} be $s_u^v(t)$, then we have:

$$s_u^v(t) = h^T Relu\left(\overline{e}_u(t) + \overline{e}_v(t) + b\right), t \in \left[t_{sub} - T, t_{sub}\right]. \tag{9}$$

Among them, h can learn a one-dimensional parameter vector, which is used to convert the Relu activation layer output into a scalar, and b is the bias factor. Then the weight of the influence of user u by the interaction items in the current time window t is:

$$s_u(t) = \sum_{v \in V} C^v s_u^v(t). \tag{10}$$

Here, C^v is the indicator variable, if user u and v have interaction within the $t \in [t_{sub} - T, t_{sub}]$ time frame $C^v = 1$, otherwise $C^v = 1$. Finally, after normalization by softmax function, the activity $a_u(t)$ of user u in the current time window t is:



(a) Recommendation accuracy (b) Recommendation coverage

Fig. 9 Selection thresholds. **a** Recommendation accuracy. **b** Recommendation coverage

$$a_u(\mathrm{t}) = softmax\left(s_u(t)\right) = \frac{exp\left(s_u(t)\right)}{\sum_{u\in G} exp\left(s_u(t)\right)}. \tag{11}$$

Group G can be divided into active subgroup $A = \left\{u|a_u(\mathrm{t}) \geq \lambda\right\}$ and inactive subgroup $NA = \left\{u|a_u(\mathrm{t}) < \lambda\right\}$ by setting the grouping threshold $\lambda = \frac{1}{k}\sum_{u\in G} a_u(t)$ according to the mean activity of all users in group G.

It is reasonable to use the mean value of the group members' activity level as the threshold for subgroup segmentation, since the mean value represents the overall activity level of the group. In this paper, we conduct several experiments on MovieLens-100K dataset and FilmTrust dataset to analyze the recommendation accuracy and recommendation coverage, and obtain the results as shown in Fig. 9. For the MovieLens-100K dataset and FilmTrust dataset, the mean values of member activity of the selected groups are 0.1 and 0.15, respectively, and the results show that when subgroups are divided using the mean value as the threshold, the highest levels of accuracy and coverage are obtained. This is because the division based on the mean value can maintain the relative balance of member activity within the group, making the activity level of members within each subgroup relatively consistent, which improves the efficiency of collaboration and the quality of communication within the subgroup. Compared with the segmentation method relying on other indicators or thresholds, the mean value-based segmentation method can better consider the overall characteristics of the group, avoiding the situation of overly relying on individual extreme values or local characteristics to make decisions. Therefore, using the mean value of activity as the segmentation threshold in group recommendation is the most effective method.

3.3.3 Subgroup Preference

Since the user is more active in the subgroup, the contribution to the consensus preference of the subgroup is also greater. Subgroup embedding is generated by weighted aggregation according to user activity weight, and subgroup preferences are obtained through the NCF model. This process integrates individual behavior and implicit influence relationship, and expands the group decision space through subgroup preference consensus, which is more conducive to the fairness of group recommendation. The embeddings of active and inactive subgroups characterize $e_A(t)$ and $e_{NA}(t)$ as follows.

$$\begin{cases} e_A(t) = \sum_{u\in A} a_u(t)\overline{e}_u(t) \\ e_{NA}(t) = \sum_{u\in NA} a_u(t)\overline{e}_u(t) \end{cases}. \tag{12}$$

Since the NCF model can deal with large-scale data and sparsity problems and can also model nonlinear relationships, this paper uses the NCF model to capture the preference similarity between subgroup members and then obtain subgroup preferences. The NCF model takes as input the subgroup embeddings $e_A(t)$ and $e_{NA}(t)$ as well as the item embeddings $\overline{e}_v(t)$, and simultaneously learns the active group–item and inactive group–item interaction rows, where the predicted score of each subgroup for the item is output and the subgroup preference is obtained accordingly. Then the prediction scores $\hat{r}_{A,v}(t)$ and $\hat{r}_{NA,v}(t)$ for item v of subgroup A and NA are as follows, where $\varnothing_1$ and $\varnothing_2$ are two NCF model parameters, respectively.

$$\begin{cases} \hat{r}_{A,v}(t) = NCF\left(e_A(t), e_v(t); \varnothing_1\right) \\ \hat{r}_{NA,V}(t) = NCF\left(e_{NA}(t), e_v(t); \varnothing_2\right) \end{cases}. \tag{13}$$

3.3.4 Model Optimization

In this paper, the regression-based pairwise ranking method is used to optimize the model, and the objective function for the active and inactive subgroups is as follows.

$$Loss_l = \sum_{(l,i,j)\in O} \left(r_{lij} - \hat{r}_{lij}\right)^2, l = AorNA, \tag{14}$$

where O represents the training set of the group recommendation task, and each interaction uses A triple (l, i, j) to represent the interacted item v_i and the uninteracted item v_j of subgroup$l = AorNA$. v_i is selected as the positive sample of subgroup$l$, and v_j is the negative sample. Let $\hat{r}_{lij} = \hat{r}_{li} - \hat{r}_{lj}$ denote the difference between the predicted scores of items that have generated interactions with the active group A and those that have not [58]. Since this paper mainly uses implicit feedback data (such as user click records) [59] for training, the true value of the rating for items that have generated interaction is 1, and the true value of the rating for items that have not generated interaction is 0, so $r_{lij} = r_{li} - r_{lj} = 1$ can be obtained. By minimizing the difference loss of $r_{lij} - \hat{r}_{lij}$, the score of subgroup l for item v_i is made close to the expected true value.

3.4 Group Recommendation

To avoid preference conflict in the process of preference fusion from subgroups to groups, the subgroup recommendation list is generated directly from the subgroup preference according to the Top-n policy, and then the subgroup recommendation list is merged into the group recommendation list. However, since the group recommendation task is also limited by the length of the Top-n list, and there are also preference conflict items in the two groups in addition to consensus preferences, it is necessary to adopt a strategy to make the best recommendation selection in the conflict items. In this paper, the average consensus aggregation (ACA) Top-n strategy (ACA_Top-n) is proposed. The subgroup $Top - c_A$ and $Top - c_{NA}$ recommendation lists are generated based on the weight ratio of subgroup average activity, and the average consensus policy is used to aggregate the subgroup recommendation list to form a group Top-n recommendation, as shown in Fig. 10.

The average consensus aggregation Top-n recommendation policy consists of three steps. 1) Obtain the length of the subgroup recommendation list. The number ratio of two groups of recommended items was calculated by the average activity of the subgroups to determine the length of

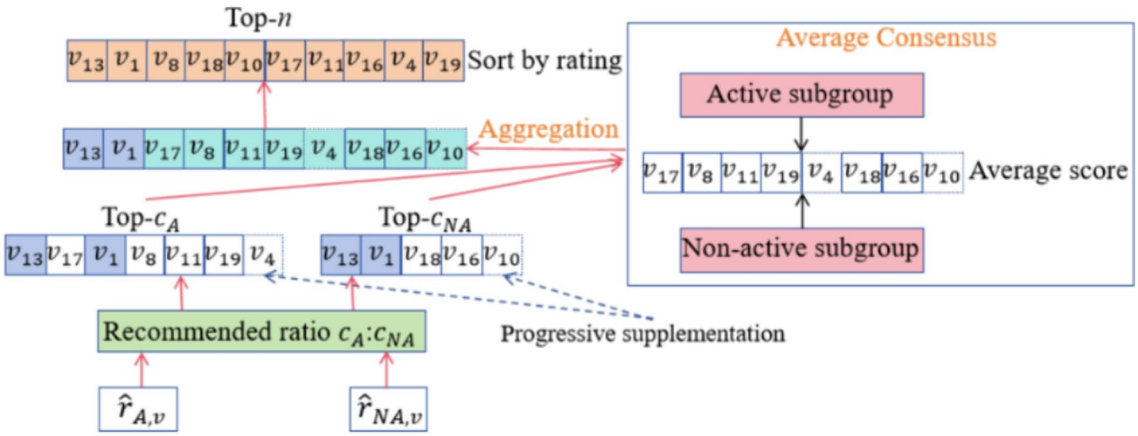


Fig. 10 Average consensus aggregation

Table 2 Recommended lists of subgroups ***A*** and ***NA***

(a) Recommendation list $\boldsymbol{R_A}$ for subgroup ***A***							
Recommended items of A	v_{13}	v_{17}	v_1	v_8	v_{11}	v_{19}	v_4
Score	0.92	0.89	0.83	0.83	0.76	0.72	0.7
(b) Recommended list $\boldsymbol{R_{NA}}$ for subgroup ***NA***							
Recommended items of NA	v_{13}	v_1	v_{18}	v_{16}	v_{10}		
Score	0.9	0.87	0.85	0.82	0.8		

the subgroup recommendation list. 2) Generate a subgroup recommendation list, sorted according to the project scores of each subgroup and selected according to the number of recommended projects. In addition to consensus projects, the projects were added to the recommendation list of the corresponding subgroup according to the recommendation ratio, so as to achieve the length of the recommendation list of each subgroup. 3) Generate a group recommendation list. Generate a consensus score for the average aggregated score of the items in both groups and generate a Top-n recommendation.

The average consensus aggregation policy combines the subgroup recommendation results to perform group recommendation. First, according to the user activity au in the subgroup, the average activity $a_A = \frac{1}{|A|}\sum_{u\in A} a_u$, of subgroups A and NA can be obtained. The subgroup with higher activity should contribute more to group recommendation. Therefore, the proportion and of each subgroup recommendation result in group recommendation can be calculated by the average activity of subgroups. If the length of the group recommendation list is set to n, the length of the recommendation list for subgroups A and NA is $c_A = [nt_A + 0.5]$, $c_{NA} = [nt_{NA} + 0.5]$, $c_A + c_{NA} = n$, and $[\bullet]$ is an integer function.

Let the recommended list of subgroups A and NA be R_A and R_{NA}, and extract x consensus preference items, then the total number of recommended items after aggregation ism $= c_A + c_{NA} - x \le n$, and $n - m$ items need to be added, $n - m$ items are still added in the two groups according to the activity weight ratio according to the score size, and then the average score of each item recommended in the two groups is calculated. Then we sort them accordingly.

Suppose $k = 10$, $c_A = 6$, $c_{NA} = 4$, the recommended list R_A and R_{NA} of subgroup A and NA (Table 2), shaded part is consensus items, there are two items in total, there are eight items in total, and two items need to be replaced. After rounding up according to the ratio of 6:4, each of the two subgroups will be replaced with one item, namely v_4 and v_{10}, respectively.

Recommended items of group G To meet the maximum consensus of all users, the average score of consensus items in the subgroup is calculated first, then there are item–score pairs (v_{13}, 0.91) and (v_1, 0.85), as shown in Table 4. For other non-consensus items, the average consensus strategy is also used for calculation. Obviously, in NA scores the recommended items in $R_A\left[v_{17}, v_8, v_{11}, v_{19}, v_4\right]$ are too low to be included in the recommended list R_{NA}, but there are still preference scores. Similarly, this phenomenon also exists for R_{NA}, as shown in Table 3. The shaded part is the project score that meets the recommended ratio of the subgroup.

The average score can be used to aggregate the consensus of active and inactive subgroups, and the Top-n group recommendation list can be obtained by combining the consensus score in order from highest to lowest (Table 4).

4 Experiment

This section first introduces FilmTrust dataset, Movielens dataset and evaluation indicators, then selects the comparison method and sets the experiment, and finally analyzes the experimental results.

4.1 Experimental Dataset

This paper uses MovieLens-100 K and FilmTrust public movie score datasets for experimental analysis. The MovieLens dataset includes information on 1,682 movie ratings from 943 users, with a total of 100,000 ratings ranging from 1 to 5. The user–movie scoring matrix is sparse, and the

Table 3 Average consensus of subgroups A and NA

Non-consensus item	v_{17}	v_8	v_{11}	v_{19}	v_4	v_{18}	v_{16}	v_{10}
Recommended items of A	0.89	0.83	0.76	0.72	0.7	0.68	0.6	0.68
Recommended items of NA	0.56	0.72	0.68	0.42	0.62	0.85	0.82	0.8
Average consensus	0.73	0.78	0.72	0.57	0.66	0.77	0.71	0.74

Table 4 Recommended list of group G

Group referral project	v_{13}	v_1	v_8	v_{18}	v_{10}	v_{17}	v_{11}	v_{16}	v_4	v_{19}
Score	0.91	0.85	0.78	0.77	0.74	0.73	0.72	0.71	0.66	0.57

Table 5 Experimental dataset

Dataset	Number of users	Number of items	Number of scores	Rating range	Sparsity
ML-100 K	943	1682	100,000	[1, 5]	93.7%
FilmTrust	1508	2071	35,497	[1, 5]	98.86%

density of the scoring matrix is 6.3%, that is, the sparsity is 93.7%. The FilmTrust dataset includes 1508 user ratings for 2071 films, for a total of 35,497 ratings on a scale of 1–5. The density of the user–movie scoring matrix is 1.14%, that is, the sparsity is 98.86%. Table 5 shows basic information about the relevant datasets.

In this paper, the two datasets are divided into the training set and the test set according to the ratio of 8:2. In the experiment, the algorithm model is trained in the training set first, and then the trained model is verified and analyzed on the test set.

4.2 Evaluation Index

4.2.1 Accuracy Rate

Recommendation accuracy is the most important measurement index in the recommendation system. In this paper, normalized loss cumulative gain is used discounted cumulative gain (nDCG) is used as the evaluation index of the accuracy of group recommendation algorithms [28]. The recommended accuracy is calculated by Eqs. (15) and (16).

$$DCG@N = \sum_{i=1}^{N} \frac{rel_i}{\log_2(i)} = rel_1 + \sum_{i=2}^{N} \frac{rel_i}{\log_2(i)}, \tag{15}$$

$$nDCG@N = \frac{DCG@N}{IDCG@N}, \tag{16}$$

where $DCG@N$ represents the satisfaction of the user with the real score of N items in the recommendation list and their relative position in the list; reli represents the score of the item ranked i in the recommendation list. $IDCG@N$ is the best-case $DCG@N$ value, where the recommendation list is ranked in descending order for the individual user.

$nDCG@N$ is obtained by the ratio of $DCG@N$ to $IDCG@N$ and ranges from 0 to 1. The higher the n $DCG@N$, the higher is the satisfaction with the recommendation list, and the better is the recommendation effect. In the experiment, the $nDCG@N$ value of each member in the group was calculated, and the $nDCG@N$ value of the group was represented by the mean value.

4.2.2 Coverage

Coverage describes the ability of a recommendation system to explore the long tail of items, which is usually calculated by the ratio of items in the user recommendation list to all items. In the group recommendation system, the coverage rate of group recommendation is also taken as the proportion of items [60]. The high coverage rate means that the coverage range of recommendation results is wide, indicating that the recommendation system has a strong ability to dig potential projects. Therefore, the coverage rate is also an evaluation index that cannot be ignored in the recommendation system. The recommended list coverage of group G is calculated as follows.

$$CoverageG = \frac{\left|rec_{genres}(G)\right|}{\left|cat_{genres}\right|}, \tag{17}$$

where it represents the number of categories included in the recommendations of group G. It indicates the amount of variety included in all projects.

4.2.3 F1 Value

F1 value is a commonly used comprehensive evaluation index to measure the performance of recommendation algorithms. It combines precision and recall to provide a more comprehensive assessment of the quality of recommendations. F1 value is calculated as the harmonic average of accuracy rate and recall rate, and the value ranges from 0 to 1. The closer it is to 1, the better is the performance of the recommendation system in balancing the accuracy rate and recall rate, that is, the recommendation results contain both high-quality recommendation items and more items of interest to users. Its advantage is that it can consider accuracy and recall rates, avoiding the limitations that may result from focusing on a single metric. The mathematical expression of $F1$ is:

$$F1 = 2 * \frac{precision * recall}{precision + recall}. \tag{18}$$

Precision refers to the percentage of recommended items that are actually of interest to the user in the recommendation results. The higher the value, the better is the prediction effect of the recommendation algorithm, which can be calculated by Eq. (19).

$$Precision = \frac{TP}{TP + FP}, \tag{19}$$

where TP (true positive) means that the project of actual interest is correctly recommended. FP (false positive) indicates that an item that is not actually of interest is wrongly recommended.

Recall refers to the percentage of items that are correctly recommended out of all items of actual interest. The higher the value, the more extensive is the recommendation of the recommendation algorithm, which can be calculated by Eq. (20).

$$Recall = \frac{TP}{TP + FN}, \tag{20}$$

Table 6 Recommendation accuracy with different number of recommendations

Algorithm	MovieLens-100 K						FilmTrust					
	5	10	15	20	25	30	5	10	15	20	25	30
FUA	0.811	0.930	0.918	0.937	0.898	0.835	0.981	0.973	0.967	0.959	0.948	0.930
AGREE	0.788	0.804	0.898	0.892	0.830	0.846	0.972	0.968	0.959	0.950	0.937	0.922
SVD_AVG	0.713	0.881	0.864	0.897	0.838	0.851	0.965	0.960	0.941	0.942	0.934	0.918
SVD_LM	0.708	0.896	0.701	0.886	0.842	0.788	0.960	0.957	0.930	0.932	0.927	0.908
SVD_MP	0.721	0.728	0.687	0.697	0.697	0.697	0.968	0.958	0.939	0.937	0.930	0.911

where *FN* (false negative) indicates a false negative, meaning that the item of actual interest is not recommended correctly.

4.3 Comparison Methods and Experimental Settings

Among the comparison methods, four comparison methods are selected to measure the effectiveness of this paper's algorithm. In the stage of individual predictive scoring (scoring matrix filling), the comparison methods are the traditional SVD (singular value decomposition) model [61]; in the stage of group scoring fusion, the comparison methods are the average-value strategy (AVG) [12], the least-misery strategy (LM) [13], and the happiest strategy (MP) [14]. To compare with the classical group recommendation algorithms, the Agree method proposed by Cao et al. [27] is also selected in this paper. In summary, the comparison methods set up in this paper are SVD_AVG, SVD_LM, SVD_MP methods based on SVD, and the Agree method. To study the influence of different algorithms on the recommendation results under different number of recommendation items, different n is selected for the experiment according to the Top-n strategy. The experiment starts with $n = 5$ and the interval is 5 until 30.

4.4 Experimental Results and Analysis

4.4.1 Recommendation Accuracy

In this experiment, normalized loss cumulative gain *nDCG* was used. The ratio of the correct data (the same data as the real data) and the recommended data in the recommendation results is used as the evaluation index of the group recommendation accuracy.

Table 6 shows the experimental results of recommendation accuracy under different number of recommendations corresponding to different group recommendation algorithms in the MovieLens and FilmTrust datasets. In this experiment, the number of recommendations n is set to 5, 10, 15, 20, 25, and 30, the time window t is randomly selected, and the group size is set to 10.

Analysis of the experimental results in Table 6 shows that on the MovieLens-100K dataset and FilmTrust dataset, the recommendation accuracy of the FUA algorithm proposed in this paper is better, indicating the effectiveness of considering the group member activity when making group recommendation and dividing into subgroups according to the member activity. For the MovieLens-100K dataset, with the increase of the group size, the recommendation accuracy shows a downward trend. The reason is that with the increase of the group size, the conflict within the group becomes more intense,

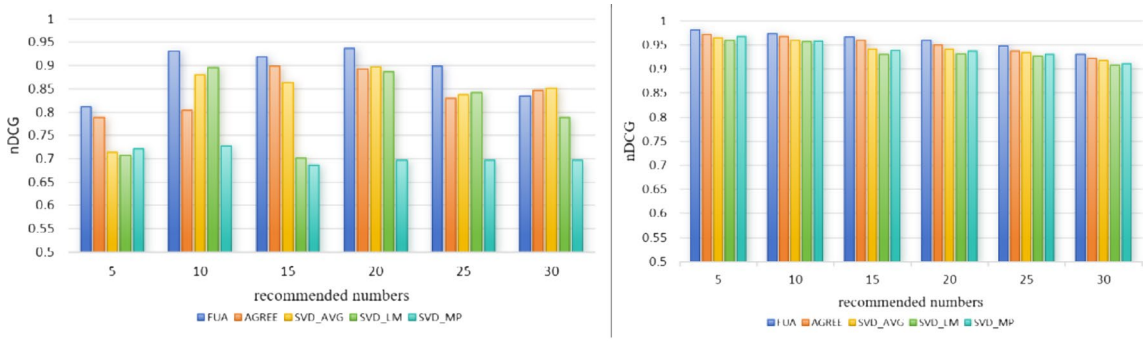


Fig. 11 Recommendation accuracy under different number of recommendations n. **a** MovieLens-100 K dataset. **b** FilmTrust dataset

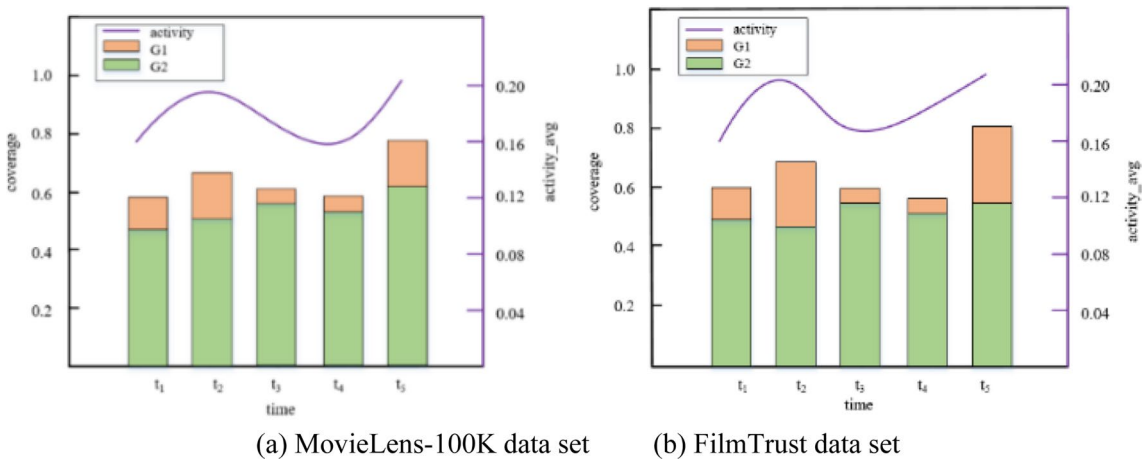


Fig. 12 Relationship between user activity and recommended coverage over time. **a** MovieLens-100 K dataset. **b** FilmTrust dataset

and the preference is more difficult to integrate, resulting in a decline in accuracy. For the FilmTrust dataset, the declining trend of recommendation accuracy shows a moderate state because the data sparsity is slightly lower than that of the MovieLens-100K dataset.

Figure 11 shows the comparison between the proposed algorithm FUA and the comparison algorithm.

On MovieLens-100 K dataset and FilmTrust dataset, the accuracy of the FUA algorithm proposed in this paper is generally higher than other baseline algorithms. First of all, the recommendation accuracy of FUA is better than the baseline algorithm because FUA alleviates the preference conflict of members in a group by dividing and fusing the recommendation list of subgroups. Secondly, with the increase of recommended items, the proportion of consensus items among group members gradually decreases, and the recommended items cannot meet the preferences of the vast majority of users, resulting in a decline in the accuracy of group recommendation. For MovieLens-100 K dataset, when the number of recommended items reaches 30, the recommendation accuracy rate is slightly lower than the AGREE algorithm and SVD_AVG, because the number of recommendations is large and the items at the end of the recommendation list cannot meet the needs of most members, so the accuracy rate decreases. For FilmTrust dataset, due to the lower sparsity of data compared with MovieLens, the declining trend of recommendation accuracy remained at a stable level.

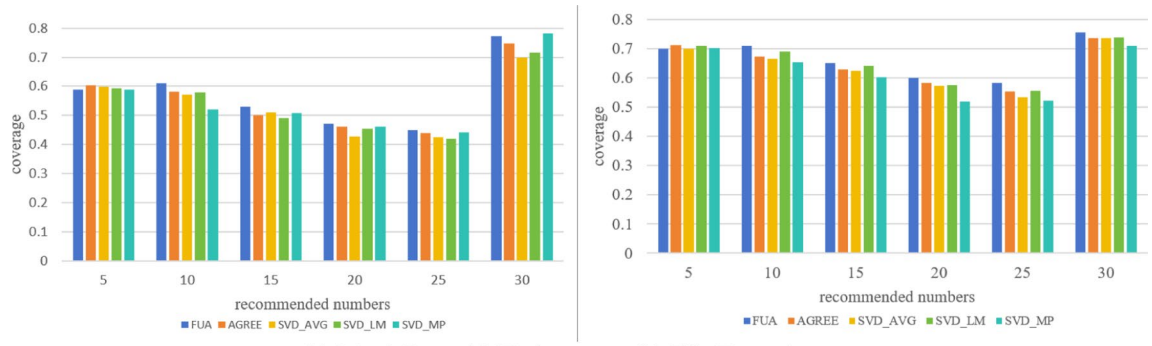
4.4.2 Recommended Coverage

Coverage is an important measure of the diversity of the recommendation system, and it can tell us whether the system recommendation can cover the entire item library, not just the popular or common items. A recommendation system with high coverage means that more items of different types and fields can be recommended, thus providing users with more comprehensive choices. Conversely, a recommendation system with low coverage may result in overly limited recommendations, with users receiving similar or duplicate recommendations. Figure 12 shows the relationship between group activity and recommendation coverage at different time t_i. The experiment selected five time node sequences in two datasets, respectively, for verification and analysis.

As can be seen from Fig. 12, the activity degree a_G of group $G1$ divided into subgroups and the recommendation coverage Gcoverage show an approximately proportional relationship Gcoverage. The more active users are, the higher is the recommendation coverage. For the group $G2$ which is not divided into groups, the recommendation

Table 7 Recommendation coverage with different number of recommendations

Algorithm	MovieLens-100 K						FilmTrust					
	5	10	15	20	25	30	5	10	15	20	25	30
FUA	0.590	0.612	0.531	0.471	0.450	0.772	0.699	0.709	0.650	0.600	0.583	0.756
AGREE	0.604	0.581	0.502	0.461	0.439	0.749	0.712	0.672	0.629	0.582	0.552	0.735
SVD_AVG	0.598	0.573	0.511	0.427	0.425	0.698	0.700	0.665	0.624	0.573	0.534	0.736
SVD_LM	0.594	0.578	0.492	0.454	0.420	0.716	0.710	0.690	0.641	0.576	0.555	0.739
SVD_MP	0.588	0.520	0.508	0.461	0.441	0.781	0.701	0.654	0.601	0.520	0.521	0.709



(a) MovieLens-100K data set (b) FilmTrust data set

Fig. 13 Recommendation coverage with different number of recommendations n. **a** MovieLens-100 K dataset. **b** FilmTrust dataset

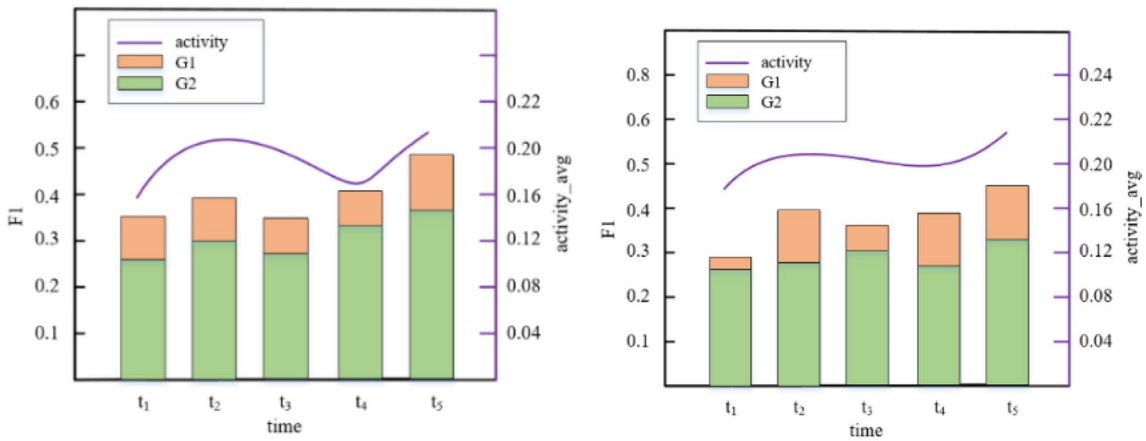
results are more inclined to active users due to relatively less attention to inactive users, which makes the recommendation coverage low.

Table 7 shows the experimental results of recommendation coverage under different numbers of recommendations corresponding to different group recommendation algorithms in MovieLens and FilmTrust datasets. In this experiment, the number of recommendations n is set to 5, 10, 15, 20, 25, and 30, the time window t is randomly selected, and the group size is set to 10.

The experimental results shown in Table 7 show that on the MovieLens-100K and FilmTrust datasets, the recommendation coverage of all algorithms shows a decreasing phenomenon, but when the number of recommended items is 30, the recommendation coverage of all algorithms increases sharply, because when the number of recommended items increases, the recommendation coverage of all algorithms increases sharply. Non-popular items get more exposure and help improve the user experience. When the number of recommended items is 5, the recommendation coverage of the FUA algorithm proposed in this paper is slightly lower than that of other algorithms. This is because when the number of recommended items is small, there are duplicate categories in the recommended items of the subgroup, resulting in fewer recommended item categories of the group as a whole, which makes the recommendation coverage low. In addition, the recommendation coverage of the proposed algorithm is generally higher than that of other algorithms, which indicates the effectiveness of the proposed algorithm for the recommendation coverage of group recommendation.

Figure 13 shows the analysis of recommendation coverage of the proposed algorithm FUA and the comparison algorithm as the number of recommendations n increases.

Locally, in Fig. 13, when the number of recommended items is 5, the recommendation coverage is slightly lower than that of the other algorithms. This is because the algorithm in this paper forms the group recommendation list by aggregating the recommendations of subgroups, and the recommended items of the two subgroups have duplicate categories, which makes the aggregated group recommendation

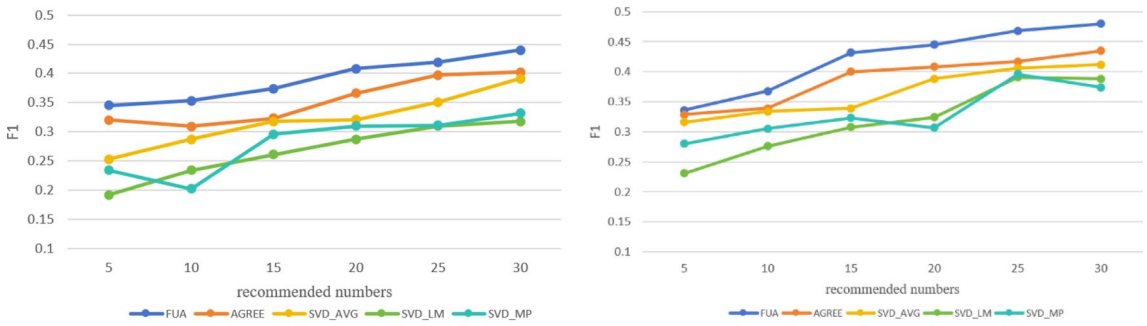


(a) MovieLens-100K data set (b) FilmTrust data set

Fig. 14 F1 values at different activity levels. **a** MovieLens-100 K dataset. **b** FilmTrust dataset

Table 8 F1 values for different number of recommendations

Algorithm	MovieLens-100 K						FilmTrust					
	5	10	15	20	25	30	5	10	15	20	25	30
FUA	0.345	0.353	0.374	0.408	0.419	0.440	0.336	0.368	0.432	0.445	0.468	0.480
AGREE	0.320	0.309	0.323	0.366	0.397	0.402	0.329	0.339	0.400	0.408	0.417	0.435
SVD_AVG	0.253	0.287	0.318	0.321	0.351	0.391	0.316	0.334	0.339	0.388	0.406	0.412
SVD_LM	0.192	0.234	0.261	0.287	0.310	0.318	0.231	0.276	0.308	0.324	0.391	0.388
SVD_MP	0.234	0.202	0.296	0.310	0.311	0.332	0.280	0.305	0.323	0.307	0.396	0.374



(a) MovieLens-100K data set (b) FilmTrust data set

Fig. 15 F1 values for different number of recommendations. **a** MovieLens-100 K dataset. **b** FilmTrust dataset

item categories less than other algorithms, resulting in relatively low recommendation coverage. However, with the increase of the length of the recommendation list, the recommended categories of the active subgroup and the inactive subgroup also gradually increase and can tend to be more complete in the group recommendation, and the recommendation coverage rate gradually increases.

On the whole, with the increase of the number of recommended items, the categories of items that meet user preferences are basically fixed, and the fluctuation is small. According to the definition of recommendation coverage (Eq. (17)), the recommendation coverage of all algorithms shows a decreasing phenomenon. However, when the number of recommendations reaches 30, the recommendation coverage rate shows a jump rise. After observation of the data, it is because when the scope of recommended items is expanded, more exposure opportunities are provided for some non-popular items, which helps to improve the recommendation novelty of users. At this time, it can better reflect the recommendation expansion advantage of the algorithm in this paper.

4.4.3 F1 Value

F1 value (Eq. (18)) considers both precision and recall, and is sensitive to the trade-off between precision and recall, which is helpful to find the performance problems of the recommendation system in these two aspects, and is a common indicator to comprehensively evaluate the performance of the recommendation system. Figure 14 shows the relationship between group activity and F1 value under t_i at different moments. The experiment selects five time node sequences in the two datasets, respectively, for verification analysis.

Table 9 Comparison of execution times (recommendations per second) between different methods

Dataset/ methods	FUA	AGREE	SVD_AVG	SVD_MP	SVD_ALM
ML-100K	27.33	36.25	53.55	58.21	54.19
FilmTrust	32.76	46.69	65.54	62.17	51.36

As can be seen from Fig. 14, on the two datasets, the change trend of F1 value and group activity is basically consistent with each other over time, and it is more similar to group $G1$ divided into subgroups. Similar to the distribution principle in Fig. 10, group $G2$, which is not divided into subgroups, pays less attention to inactive users, and its recommendation results are more inclined to active users, so that the F1 value of the group is lower than $G1$ under the same activity degree.

Table 8 shows the experimental results of F1 values under different numbers of recommendations corresponding to different group recommendation algorithms in MovieLens and FilmTrust datasets. In this experiment, the number of recommendations n is set to 5, 10, 15, 20, 25, and 30, the

time window t is randomly selected, and the group size is set to 10.

The experimental results shown in Table 8 show that on the MovieLens-100K and FilmTrust datasets, the FUA algorithm proposed in this paper generally has higher F1 value than other algorithms, and the F1 value gradually increases with the increase of the number of recommended items. On the FilmTrust dataset, when the number of recommended items is small, the F1 value advantage of the proposed algorithm is smaller than that of other baseline algorithms, but with the increase of the number of recommended items, the advantage of the proposed algorithm is more obvious. This is because when the number of recommended items is small, the categories of group-recommended items of different algorithms are almost the same, and there is little difference in F1 value between the proposed algorithm and other baseline algorithms. However, with the increase of the number of recommended items, the recommended items can meet the needs of users, and the proposed algorithm can improve the precision and recall rate of members.

Figure 15 shows the trend analysis of the proposed algorithm FUA and the comparison algorithm with the increasing F1 value of the number of recommendations n.

Because the F1 value will have different emphasis on different project categories, it may cause some categories to be ignored in the evaluation. This paper broadens the coverage of project categories by dividing into subgroups, which effectively alleviates the limitations of the F1 value, and makes the use of F1 value evaluation better reflect the advantages of the algorithm in this paper. Figure 15 shows that the F1 value of the proposed FUA method on the two datasets is higher than that of other comparison algorithms, and with the increase of the number of recommended items, the F1 value shows a gradual increasing trend, and the F1 value ranking is FUA > AGREE > SVD_AVG > SVD_MP > SVD_LM.

4.4.4 Execution Time

To demonstrate the success of the proposed method in this paper in improving the non-scalability problem, we compare the proposed method with other methods in terms of recommendation speed and the results are shown in Table 9. The execution time of this paper's method on the first and second dataset is 24.99 and 35.28 s. The results show that our method completes the recommendation in a shorter time than the other methods.

5 Conclusion

In this paper, we propose a group recommendation algorithm based on FUA. Firstly, combined with NMF and K-means method, GDPCP algorithm was proposed to discover groups of user–item rating matrix. Secondly, according to the user–item interaction bipartite graph, the deep neural network and self-attention mechanism are used for graph representation learning to obtain the high-order feature representations of users and items. Then, on the basis of designing a sliding time window, the activity of each user in the group was calculated through the attention network, and the group was divided into active and inactive subgroups according to the threshold. According to the subgroup embedding, the interaction function of active subgroup–item and inactive subgroup–item was learned through the NCF network to obtain the preferences of each subgroup. Finally, according to the subgroup preferences, the candidate recommendation lists were generated for the active subgroup and the inactive subgroup according to the Top-n rule, and the two candidate recommendation lists were aggregated according to the ACA_Top-n method to generate the final recommendation results of the group, and a more fair group recommendation model was established. The experimental results on MovieLens dataset and FilmTrust dataset show the performance advantages of the proposed algorithm.

Although this study takes into account the preference differences of members in the group, which effectively improves the fairness of recommendation. However, there are some limitations, for example, the influence of user's social network on the recommendation performance is not considered, and the social network contains valuable information about the trust relationship between users and how users form groups. How to explore the social connections of group members to further improve the fairness of group recommendation will be the next research goal.

Authors' Contributions JJ was responsible for providing the research direction, guiding the experimental design, and critically revised the manuscript. FW conceived and designed the study, provided technical support and performed experiments, analyzed data and interpretation, and prepared the manuscript. HW participated in data collection and data analysis. SL participated in the literature review and manuscript preparation. All authors read and approved the final manuscript.

Funding 2023 Natural Science Foundation of Gansu Province (23JRRA686).

Data Availability Research data are not shared.

Declarations

Conflicts of interests The authors declare that they have no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Shrivastava, R., Sisodia, D.S., Nagwani, N.K.: Deep neural network-based multi-stakeholder recommendation system exploiting multi-criteria ratings for preference learning. Expert Syst. Appl. **213**, 119071 (2023)
2. Breese, J. S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. arXiv preprint arXiv.1301.7363 (2013).
3. Procof the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 447–456 (2009).
4. Ma H, Yang H, Lyu M R, et al.: Sorec: social recommendation using probabilistic matrix factorization. Proceedings of the 17th ACM conference on Information and knowledge management. Napa Valley, USA: ACM. 931–940 (2008).
5. Jia, J., Yao, Y., Chen, W.: Group recommendation algorithm based on non-negative matrix decomposition. Comput. Eng. Sci. **44**(05), 933–943 (2022)
6. Lian, J., Zhang, F., Xie, X., Sun, G.: Towards Better Representation Learning for Personalized News Recommendation: a Multi-Channel Deep Fusion Approach. In IJCAI pp. 3805–3811 (2018).
7. Jia, J., Chen, S., Shang, T.: A group recommendation algorithm based on dividing subgroup. Adv Theory Simul **5**(12), 2200557 (2022)
8. Blumer, H.: Social problems as collective behavior. Soc. Probl. **18**(3), 298–306 (1971)
9. Back, K.W.: Influence through social communication. J. Abnorm. Soc. Psychol. **46**(1), 9 (1951)
10. Page, S.: The difference: How the power of diversity creates better groups, firms, schools, and societies-new edition. Princeton University Press (2008).
11. Mansoury, M., Abdollahpouri, H., Pechenizkiy, M., Mobasher, B.,Burke, R.: Feedback loop and bias amplification in recommender systems. In Proceedings of the 29th ACM international conference on information & knowledge management.pp. 2145–2148 (2020).
12. Yera, R., Alzahrani, A.A., Martínez, L.: A fuzzy content-based group recommender system with dynamic selection of the aggregation functions. Int. J. Approximate Reasoning **150**, 273–296 (2022)
13. Pérez-Almaguer, Y., Yera, R., Alzahrani, A.A., Martínez, L.: Content-based group recommender systems: A general taxonomy and further improvements. Expert Syst. Appl. **184**, 115444 (2021)
14. Boratto, L., Carta, S., Fenu, G.: Discovery and representation of the preferences of automatically detected groups: Exploiting the link between group modeling and clustering. Futur. Gener. Comput. Syst. **64**, 165–174 (2016)
15. Nozari, R.B., Koohi, H.: A novel group recommender system based on members' influence and leader impact. Knowledge-Based Syst. **205**(1), 106296 (2020)
16. Masthoff, J.: Group modeling: Selecting a sequence of television items to suit a group of viewers. Personalized Digital Television: Targeting Programs to individual Viewers, 93–141 (2004).
17. Christensen, I.A., Schiaffino, S.: Entertainment recommender systems for group of users. Expert Syst. Appl. **38**(11), 14127–14135 (2011)
18. Yalcin, E., Ismailoglu, F., Bilge, A.: An entropy empowered hybridized aggregation technique for group recommender systems. Expert Syst. Appl. **166**, 114111 (2021)
19. Ji, J., Pei, J., Lin, S, et al.: Multi-granularity attention model for group recommendation. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. pp. 3973–3977 (2023).
20. Yujie, Z.H.A.N.G., Yulu, D.U., Xiangwu, M.E.N.G.: Research on group recommender systems and their applications. Chinese J. Comp. **39**(4), 745–764 (2016)
21. Garcia, I., Pajares, S., Sebastia, L., Onaindia, E.: Preference elicitation techniques for group recommender systems. Inf. Sci. **189**, 155–175 (2012)
22. Chen, Y.L., Cheng, L.C., Chuang, C.N.: A group recommendation system with consideration of interactions among group members. Expert Syst. Appl. **34**(3), 2082–2090 (2008)
23. Masthoff, J.: Recommender Systems Handbook. Springer-Verlag, New York (2010)
24. Liu, X., Tian, Y., Ye, M., Lee, W. C.: Exploring personal impact for group recommendation. In Proceedings of the 21st ACM international conference on Information and knowledge management. pp. 674–683 (2012).
25. Yuan, Q., Cong, G., Lin, C. Y.: COM: a generative model for group recommendation. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 163–172 (2014).
26. Kaya, M., Bridge, D., Tintarev, N.: Ensuring fairness in group recommendations by rank-sensitive balancing of relevance. In Proceedings of the 14th ACM Conference on Recommender Systems. pp. 101–110 (2020).
27. Cao, D., He, X., Miao, L., An, Y., Yang, C., Hong, R.: Attentive group recommendation. In The 41st International ACM SIGIR conference on research & development in information retrieval. pp. 645–654 (2018).
28. Yin, H., Wang, Q., Zheng, K., Li, Z., Yang, J., & Zhou, X.: Social influence-based group representation learning for group recommendation. In 2019 IEEE 35th International Conference on Data Engineering (ICDE). pp. 566–577 (2019).
29. Felfernig, A., Boratto, L., Stettinger, M., Tkalčič, M. (eds.): Group recommender systems: An introduction. Springer, Cham (2024)
30. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Philip, S.Y.: A comprehensive survey on graph neural networks. IEEE Trans. Neural Netw. Learn. Syst. **32**(1), 4–24 (2020)
31. Ye, S., Liang, J., Liu, R., Zhu, X.: Symmetrical graph neural network for quantum chemistry with dual real and momenta space. J. Phys. Chem. A **124**(34), 6945–6953 (2020)
32. Wang, Z., Yang, F., Xu, Q., Wang, Y., Yan, H., Xie, M.: Capacity estimation of lithium-ion batteries based on data aggregation and feature fusion via graph neural network. Appl. Energy **336**, 120808 (2023)
33. Chen, X., Jia, S., Xiang, Y.: A review: Knowledge reasoning over knowledge graph. Expert Syst. Appl. **141**, 112948 (2020)
34. Yao, L., Mao, C., Luo, Y.: Graph convolutional networks for text classification. In Proceedings of the AAAI conference on artificial intelligence. Vol. 33, No. 01, pp. 7370–7377 (2019).
35. Wu, L., Sun, P., Hong, R., Fu, Y., Wang, X., Wang, M.: Socialgcn: An efficient graph convolutional network based model

for social recommendation. arXiv preprint arXiv:1811.02815 (2018).
36. Peng, H., Wang, H., Du, B., et al.: Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting. Inf. Sci. **521**, 277–290 (2020)
37. Jiang, W., Luo, J.: Graph neural network for traffic forecasting: A survey. Expert Syst. Appl. **207**, 117921 (2022)
38. Li, T., Zhou, Z., Li, S., Sun, C., Yan, R., Chen, X.: The emerging graph neural networks for intelligent fault diagnostics and prognostics: A guideline and a benchmark study. Mech. Syst. Signal Process. **168**, 108653 (2022)
39. Wang, J., Liu, X., Shen, S., Deng, L., Liu, H.: DeepDDS: deep graph neural network with attention mechanism to predict synergistic drug combinations. Briefings in Bioinformatics, 23(1), bbab390 (2022).
40. Venkatapathy, S., Votinov, M., Wagels, L., Kim, S., et al.: Ensemble graph neural network model for classification of major depressive disorder using whole-brain functional connectivity. Front. Psych. **14**, 1125339 (2023)
41. Ying, R., He, R., Chen, K., Eksombatchai, P., et al.: Graph convolutional neural networks for web-scale recommender systems. In Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 974–983 (2018).
42. Sun, J., Zhang, Y., Ma, C., Coates, M., et al.: Multi-graph convolution collaborative filtering. In 2019 IEEE international conference on data mining (ICDM). pp. 1306–1311 (2019).
43. S Sun, J., Zhang, Y., Guo, W., Guo, H., et al.: Neighbor interaction aware graph convolution networks for recommendation. In Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval. pp. 1289–1298 (2020).
44. Pfadler, A., Zhao, H., Wang, J., Wang, L., et al.: Billion-scale recommendation with heterogeneous side information at taobao. In 2020 IEEE 36th International Conference on Data Engineering (ICDE). pp. 1667–1676 (2020).
45. Chang, J., Gao, C., Zheng, Y., Hui, Y., et al.: Sequential recommendation with graph neural networks. In Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval. pp. 378–387 (2021).
46. Wu, J., He, X., Wang, X., Wang, Q., et al.: Graph convolution machine for context-aware recommender system. Front. Comp. Sci. **16**(6), 166614 (2022)
47. Liu, C., Li, Y., Lin, H., Zhang, C.: GNNRec: gated graph neural network for session-based social recommendation model. J. Intell. Inform. Syst. **60**(1), 137–156 (2023)
48. Chen, J., Li, H., Zhang, X., Zhang, F., et al.: SR-HetGNN: session-based recommendation with heterogeneous graph neural network. Knowl. Inf. Syst. **66**(2), 1111–1134 (2024)
49. Yu, L., Leng, Y., Zhang, D., He, S.: Collaborative group embedding and decision aggregation based on attentive influence of individual members: A group recommendation perspective. Decis. Support. Syst. **165**, 113894 (2023)
50. He, X., Deng, K., Wang, X., Li, Y., et al.: Lightgcn: Simplifying and powering graph convolution network for recommendation. In Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval. pp. 639–648(2020)
51. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., et al.: Attention is all you need. Advances in neural information processing systems, 30 (2017)
52. Wang, H.L., Yang, D., Nie, T.Z., Kou, Y.: Attributed heterogeneous information network embedding with self-attention mechanism for product recommendation. J. Comput. Res. Dev **59**(7), 1509–1521 (2022)
53. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473(2014)
54. Zhang, H., Goodfellow, I., Metaxas, D., Odena, A.: Self-attention generative adversarial networks. In International conference on machine learning. pp. 7354–7363 (2019)
55. He, X., Liao, L., Zhang, H., Nie, L.: Neural collaborative filtering. In Proceedings of the 26th international conference on world wide web. pp. 173–182 (2017)
56. Liu, H., Ren, H., He, H.: Group recommendation menthod based on self-attention mechanism. Appl. Res. Comp. **37**(12), 3572–3577 (2020)
57. Yin, H., Wang, Q., Zheng, K., Li, Z., et al.: Social influence-based group representation learning for group recommendation. In 2019 IEEE 35th International Conference on Data Engineering (ICDE). pp. 566–577 (2019)
58. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L. BPR: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618(2012)
59. Wang, X., He, X., Nie, L., Chua, T. S. Item silk road: Recommending items from information domains to social users. In Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval. pp. 185–194 (2017)
60. Hu, C., Meng, X., Zhang, Y., Du, Y.: Enhanced group recommendation method based on preference aggregation. J. Softw. **29**(10), 3164–3183 (2017)
61. Yuan, X., Han, L., Qian, S., Xu, G., et al.: Singular value decomposition based recommendation using imputed data. Knowl.-Based Syst. **163**, 485–494 (2019)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.